Semantic Web 15 (2024) 959–989
DOI 10.3233/SW-233282
IOS Press

Explanation Ontology: A general-purpose, semantic representation for supporting user-centered explanations

Shruthi Chari [a,*], Oshani Seneviratne [a], Mohamed Ghalwash [b], Sola Shirai [a], Daniel M. Gruen [a], Pablo Meyer [b], Prithwish Chakraborty [b] and Deborah L. McGuinness [a]

[a] *Computer Science, Rensselaer Polytechnic Institute, NY, US*
E-mails: charis@rpi.edu, senevo@rpi.edu, shiras2@rpi.edu, gruend2@rpi.edu, dlm@cs.rpi.edu
[b] *Center for Computational Health, IBM Research, NY, US*
E-mails: mohamed.ghalwash@ibm.com, pmeyerr@us.ibm.com, prithwish.chakraborty@ibm.com

Editors: Roberto Confalonieri, University of Padua, Italy; Oliver Kutz, Free University of Bozen-Bolzano, Italy; Diego Calvanese, Umeå University, Sweden and Free University of Bozen-Bolzano, Italy; Jose M. Alonso, University of Santiago de Compostela, CiTIUS, Spain; Shang-Ming Zhou, University of Plymouth, UK
Solicited reviews: Ilaria Tiddi, Vrije Universiteit Amsterdam, Netherlands; two anonymous reviewer

Abstract. In the past decade, trustworthy Artificial Intelligence (AI) has emerged as a focus for the AI community to ensure better adoption of AI models, and explainable AI is a cornerstone in this area. Over the years, the focus has shifted from building transparent AI methods to making recommendations on how to make black-box or opaque machine learning models and their results more understandable by experts and non-expert users. In our previous work, to address the goal of supporting user-centered explanations that make model recommendations more explainable, we developed an Explanation Ontology (EO). The EO is a general-purpose representation that was designed to help system designers connect explanations to their underlying data and knowledge. This paper addresses the apparent need for improved interoperability to support a wider range of use cases. We expand the EO, mainly in the system attributes contributing to explanations, by introducing new classes and properties to support a broader range of state-of-the-art explainer models. We present the expanded ontology model, highlighting the classes and properties that are important to model a larger set of *fifteen* literature-backed explanation types that are supported within the expanded EO. We build on these explanation type descriptions to show how to utilize the EO model to represent explanations in *five* use cases spanning the domains of finance, food, and healthcare. We include competency questions that evaluate the EO's capabilities to provide guidance for system designers on how to apply our ontology to their own use cases. This guidance includes allowing system designers to query the EO directly and providing them exemplar queries to explore content in the EO represented use cases. We have released this significantly expanded version of the Explanation Ontology at https://purl.org/heals/eo and updated our resource website, https://tetherless-world.github.io/explanation-ontology, with supporting documentation. Overall, through the EO model, we aim to help system designers be better informed about explanations and support these explanations that can be composed, given their systems' outputs from various AI models, including a mix of machine learning, logical and explainer models, and different types of data and knowledge available to their systems.

Keywords: Explainable AI, semantic representation of explanations, Explanation Ontology, modeling explanation types – AI method outputs and knowledge, supporting patterns for explanation types

[*] Corresponding author. E-mail: charis@rpi.edu.

1. Introduction

The uptake in the use of Artificial Intelligence (AI) models, and machine learning (ML) models in particular, has driven a rise in awareness and focus in explainable and interpretable AI models [21,27]. The diverse requirements around explainability point to the need for computational solutions to connect different components of AI systems, including method outputs, user requirements, data, and knowledge, to compose user-centered explanations, both for system designers and the consumers of the system, or, end-users or users in general. These requirements present opportunities that align well with the strengths of semantic technologies, particularly ontologies and knowledge graphs (KGs), that can represent entities and relationships between them for either reasoning or querying to support upstream tasks. A semantic structuring of the explanation space and its contributing attributes can help system designers support the requirements for explanations more efficiently (see Section 2 and Section 4 for some examples).

Previously, we proposed and developed an Explanation Ontology (EO) [13] that provides a semantic structuring for the entities that contribute to explanations from a user, interface, and system perspective. However, with the ever evolving explainable AI research landscape, we found value in expanding the EO model to include additional requirements and support a broader range of use cases. The additional requirements were partially motivated by papers from several researchers [18,34] who posit that different classes of users have different requirements for assistance from AI systems. These translate to further questions that users want answered. Examples include additional evidence over model explanations and decisions that provide scores for domain experts and more data-centric explanations of the dataset and method capabilities for data scientists. Further, several taxonomies of machine learning explainability methods [2,3,64] offer guides for how different classes of methods can support model-centric classifications of explanations. However, what is lacking is a tool for system designers, who build and design explainable interfaces from the method outputs, to develop these method outputs to explanations that can address their user questions. Hence, within additions to the EO, we capture the capabilities of various AI methods including explanation methods and their ability to support literature-derived explanation types. We aim to support system designers who through the use of the EO can determine what explanations can be supported from the method outputs, datasets and knowledge stores at their disposal and plan for explanations that their use case participants would require.

In this paper, we describe the expanded EO model,[1] a general-purpose semantic representation for explanations, in detail. We first outline the core EO model that includes the main classes and properties that users need to model their explanations. Then we demonstrate how the EO model can be used to represent fifteen literature-derived explanation types (six of these fifteen types are introduced in the expanded EO model). We then describe how the ontology can be used to represent and infer different explanation types in various AI use cases where model outputs are already present. With these goals, we divide the paper as follows. First, in Section 2, we motivate the need for the expanded EO model to represent user-centered explanation types and introduce a running example that illustrates the need for a unified semantic representation for explanations and their dependencies. Then, in Section 3, we present the expanded EO model, with an emphasis on the main classes and properties, a deeper-dive into the AI method and explanation branches, and a description of how the EO model is used to represent sufficiency conditions for the literature-derived explanation types that we support. With this background of the EO model, in Section 4, we show that the EO can represent outputs in *five* different use cases spanning domains including food, healthcare, and finance, and also show how these representations allow explanations to be classified into the EO's literature-derived explanation types. Finally, in Section 6, we evaluate the EO using two approaches. First, a task-based approach, that evaluates the EO's ability to answer a representative set of competency questions that a system designer would want addressed to familiarize themselves with the EO before they apply it to their use cases. Second, we evaluate from a coverage perspective, the types and breadth of content about explanations that can be queried from our EO represented use cases. Finally, in Section 7, we present an overview of the related work where semantic representations of explanations have been attempted, and describe how they either do not use standard formats, or do not capture all the components that explanations are dependent on. Overall, we expect this paper and its associated website resource (https://tetherless-world.github.io/explanation-ontology) to serve as a primer for system designers and other interested individuals from two perspectives. First when system designers are

[1] In the rest of the paper, we refer to the expanded EO model as EO. When we compare the current model to the previous version, we make an explicit reference to the previous EO version, v1.0, via a citation [13] or previous EO phrasing.

considering types of explanations to support, they may review our explanation types and the motivating needs for those types. Second, when system designers decide to apply the EO directly to address their needs for user-centered explanations in settings that require the combination of different model outputs and content from diverse data and knowledge sources.

2. Background

2.1. Motivation: Why are user-centered explanations necessary?

In recent years, with principles introduced in global policy around AI such as in Europe's General Data Protection Regulation (GDPR) act[2] and the White House's National AI Research Resource (NAIRR) Task Force,[3] there has been a growing focus on trustworthy AI. This focus has reflected the need for transparency and trust around the vast amounts of data collected by parties in multiple domains and has brought to light potential concerns with the AI models increasingly used on such data to affect important decisions and policy. In the trustworthy AI age, several position statements [21,27,40,41] focused on directions to move towards explainable, fair, and causal AI. These papers inspired computational efforts to improve trust in AI models. For example, to enable explainability in composition to provide confidence in model usage, IBM released the AIX-360 toolkit [2,3], with multiple explainer methods capable of providing different types of model explanations, including local, global, and post-hoc explanations. At the same time, there have been user studies [13,27,36,63] on the explanation types that users require and the questions that they want to be addressed, illustrating that user-centered explainability is question-driven, conversational and contextual.

Inspired by these studies, we reviewed the social sciences and computer science literature [11] regarding explanations and cataloged several user-centered explanation types that address different types of user questions and the associated informational needs. We conducted expert panel studies with clinicians [28] to understand which of these explanation types were most helpful in guideline-based care. We found that clinicians prefer a combination of holistic, scientific, and question-driven explanations connected to broader medical knowledge and their implications in context, beyond typical model explanations which focus only on the specific data and AI mechanisms used. In line with our findings, Dey et al. [19] recently illustrated a spectrum of clinical personas and their diverse needs for AI explainability. Also, recent papers point out that current model explainability [25] does not align with the human-comprehensible explanations that different domain experts expect when using AI aids in their practice. This reinforces the idea that explanations in real-world applications need to involve domain knowledge and be sensitive to the prior knowledge, usage situations, and resulting informational needs of the users to whom they are delivered.

These studies point to still unmet needs in bridging the gap between explanations that AI models can generate and what users want. This gap motivated us to design the EO to represent user-centered explanation types and their dependencies. We aimed to provide system designers with a single resource that could be used to identify critical user-centered explanation requirements, and further to provide building blocks for them to use in their designs. In our modeling of the EO (Section 3), we take into account our learnings from our literature review for various explanation types and include terminology that is typically associated with explanations both from the perspective of end-users (e.g., from the clinicians we interviewed) that consume them, and the AI methods that generate them.

2.2. Illustrative example

We are developing methods to support user-centered explanations that provide context around entities of interest in risk prediction settings, involving the chances of individual patients developing comorbidities of a chronic disease such as type-2 diabetes [10]. In a risk-prediction use case, clinicians and researchers such as data scientists consult

[2] GDPR: https://gdpr.eu

[3] White House AI Task Force: https://www.whitehouse.gov/ostp/news-updates/2022/05/25/bridging-the-resource-divide-for-artificial-intelligence-research/.

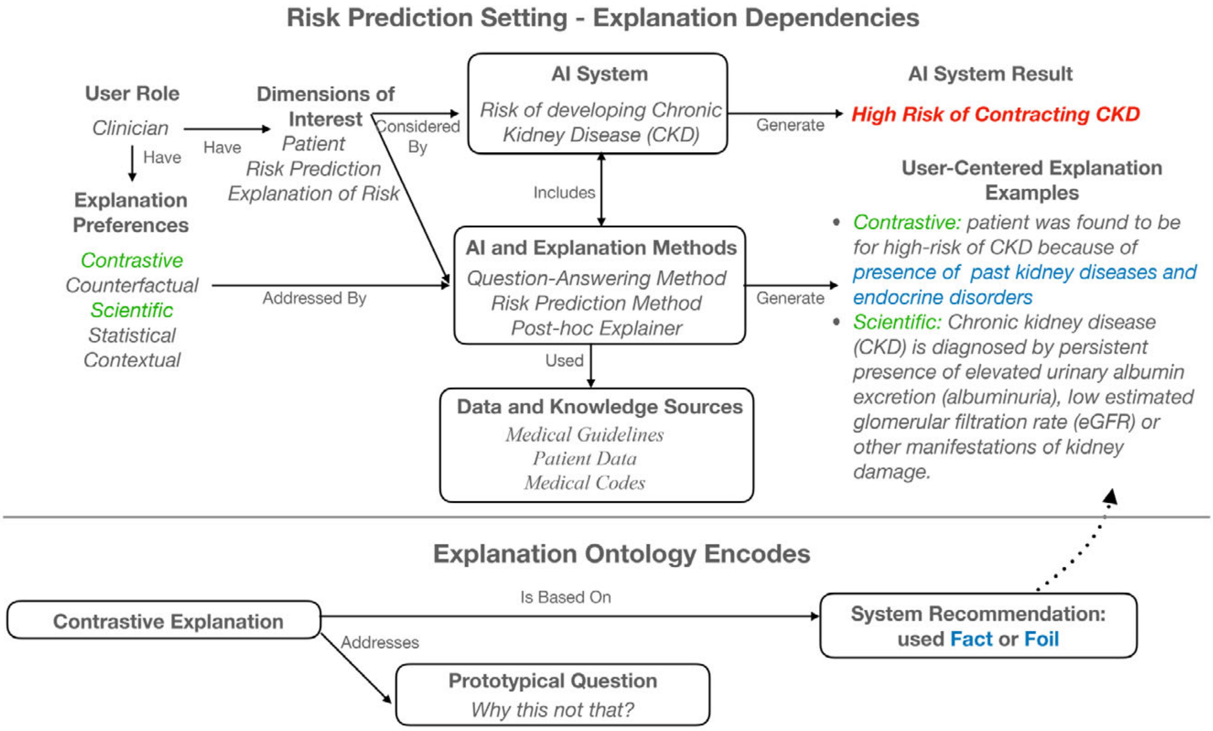

Risk Prediction Setting - Explanation Dependencies
User Role
Clinician
Have
Dimensions of Interest
Patient
Risk Prediction
Explanation of Risk
Have
Considered By
Explanation Preferences
Contrastive
Counterfactual
Scientific
Statistical
Contextual
Addressed By
AI System
Risk of developing Chronic Kidney Disease (CKD)
Generate
AI System Result
High Risk of Contracting CKD
Includes
AI and Explanation Methods
Question-Answering Method
Risk Prediction Method
Post-hoc Explainer
Generate
Used
Data and Knowledge Sources
Medical Guidelines
Patient Data
Medical Codes
User-Centered Explanation Examples
Contrastive: patient was found to be for high-risk of CKD because of presence of past kidney diseases and endocrine disorders
Scientific: Chronic kidney disease (CKD) is diagnosed by persistent presence of elevated urinary albumin excretion (albuminuria), low estimated glomerular filtration rate (eGFR) or other manifestations of kidney damage.
Explanation Ontology Encodes
Contrastive Explanation
Is Based On
System Recommendation: used Fact or Foil
Addresses
Prototypical Question
Why this not that?

Fig. 1. Illustrative examples of explanations clinicians were looking for in our risk prediction setting and how these explanations need to be supported from different data sources and AI methods [10]. Seen here is also a template of a contrastive explanation, a type of user-centered explanation supported within the Explanation Ontology, and how the configuration of this class can allow system designers to plan for how to support facts that contrastive explanations are based on.

different data and knowledge sources and use insights from multiple reasoning methods during their decision-making. In this section, we will focus on findings from our work to support the explainability needs of our target users and use these findings to inform the kinds of explanations the EO should support and illustrate how a semantic approach such as ours would be helpful to system designers who build applications in similar evolving ecosystems.

In their decision-making, clinicians interact with multiple information sources and views, involving reasoning and data associated with different explanations types. These include contrastive explanations as they choose between alternative treatments, contextual ones to understand more deeply the implications of a treatment given the patient's risk and history, and scientific explanations that situate results in terms of current literature. As has been identified in explainable AI literature [36], each of these user-centered explanation types can be seen to address a certain kind of question. These questions are each capable of being supported by different AI method(s) in interaction with data and knowledge sources. For example, in our risk prediction setting, we identified three dimensions of context (Fig. 1) around which to provide answers. The identification of these dimensions let us devise a set of five question types that clinicians would require to be addressed by explanations [10].

In summary, when developing AI-enabled tools to assist their end-users in real world use cases, system designers can benefit from an infrastructure that helps them identify potential explanations their end-users may need, and map them to the technological methods and data needed to provide those explanations. The EO is a semantic representation that can help system designers understand the various kinds of explanations that may be desired, and determine what capabilities their systems would need to include to support those explanations. The EO can also serve as the basis of a registry system that maps between explanation needs and an evolving set of computational services and methods that can provide such explanations, enabling systems to be constructed that will provide the best currently available explanations without the need for significant rewriting.

An example of an explanation built from its dependencies is seen in the user-centered explanation examples portion of Fig. 1, wherein a scientific explanation is populated from the output of an AI-enabled question-answering

Table 1

List of ontology prefixes used in the paper and its images, tables and listings

Ontology Prefix	Ontology	URI
sio	SemanticScience Integrated Ontology	http://semanticscience.org/resource/
prov	Provenance Ontology	http://www.w3.org/ns/prov-o#
eo	Explanation Ontology	https://purl.org/heals/eo#
ep	Explanation Patterns Ontology	http://linkedu.eu/dedalo/explanationPattern.owl#

method that is run on type-2 diabetes guideline literature. In such a case, the EO signals to the system designer that scientific explanations would require a literature source, and if such content is not readily accessible, they would need to determine how to extract this content. Hence, the EO can be a useful tool for system designers to plan for and structure support for explanations in their systems. These ideas will become more apparent when we go through the EO model in Section 3 and provide examples of how it is applied to structure explanations in various use cases in Section 4.

3. Explanation Ontology

We designed the Explanation Ontology (EO) (both the original [13] and the current expanded version) to model attributes related to explanations that are built upon records available to AI systems such as datasets, knowledge stores, user requirements, and outputs of AI methods. In the EO, we capture attributes that would allow for the generation of such user-centered explanations from various components that the system has at its disposal. We also provide templates that utilize the EO model to structure different literature-derived explanation types that each address various prototypical question types that users might have [36]. Here, we describe the details of the EO's core class and property model (Section 3.1) and the requirements and modeling of the *fifteen* different user-centered explanation types that we support (Section 3.2). The EO model is general-purpose and can be extended at its core structure to represent explanations in use cases spanning different domains (Section 4).

3.1. Ontology composition

For the core EO model, we adopted a bottom-up approach to narrow down the classes about the central 'explanation'[4] concept by analyzing what terms are most often associated with explanations in explanation method papers and position statements that describe a need for various explanation types. We found that the terms that most often described explanations included attributes that they directly interact with or are generated by, including the 'AI method' generating the explanation, the 'user' consuming it, and other interface attributes such as the user 'question' that the explanation addresses. We crystallized our understanding of the relationships between these terms in the core EO model as shown in Fig. 2. We adopted a top-down approach to expand beyond the core EO model by refining, editing, and adding to this model when representing explanation types and more specific sub-categorizations to certain classes, such as the 'AI method.'

In the EO, we capture the attributes that explanations build upon, such as a 'system recommendation' and 'knowledge,' and model their interactions with the 'question' (s) they address. We can broadly categorize explanation dependencies into the system, user, and interface attributes as seen in Fig. 2. These categorizations help us cover a broad space of attributes that explanations directly or indirectly interact with.

User Attributes: User attributes are the concepts that are related to a 'user' who is consuming an explanation. These include the 'question' that the user is asking, 'user characteristic's that describe the user, and the user's 'situation'. Explanations that the user is consuming are modeled to address the user's 'question,' and may also take into account factors such as the user's 'situation' or their 'user characteristic' such as their education and location.

[4] In this section, we refer to ontology classes by their labels in single quotes. The labels also correspond to the ontology class identifiers themselves (IRIs).

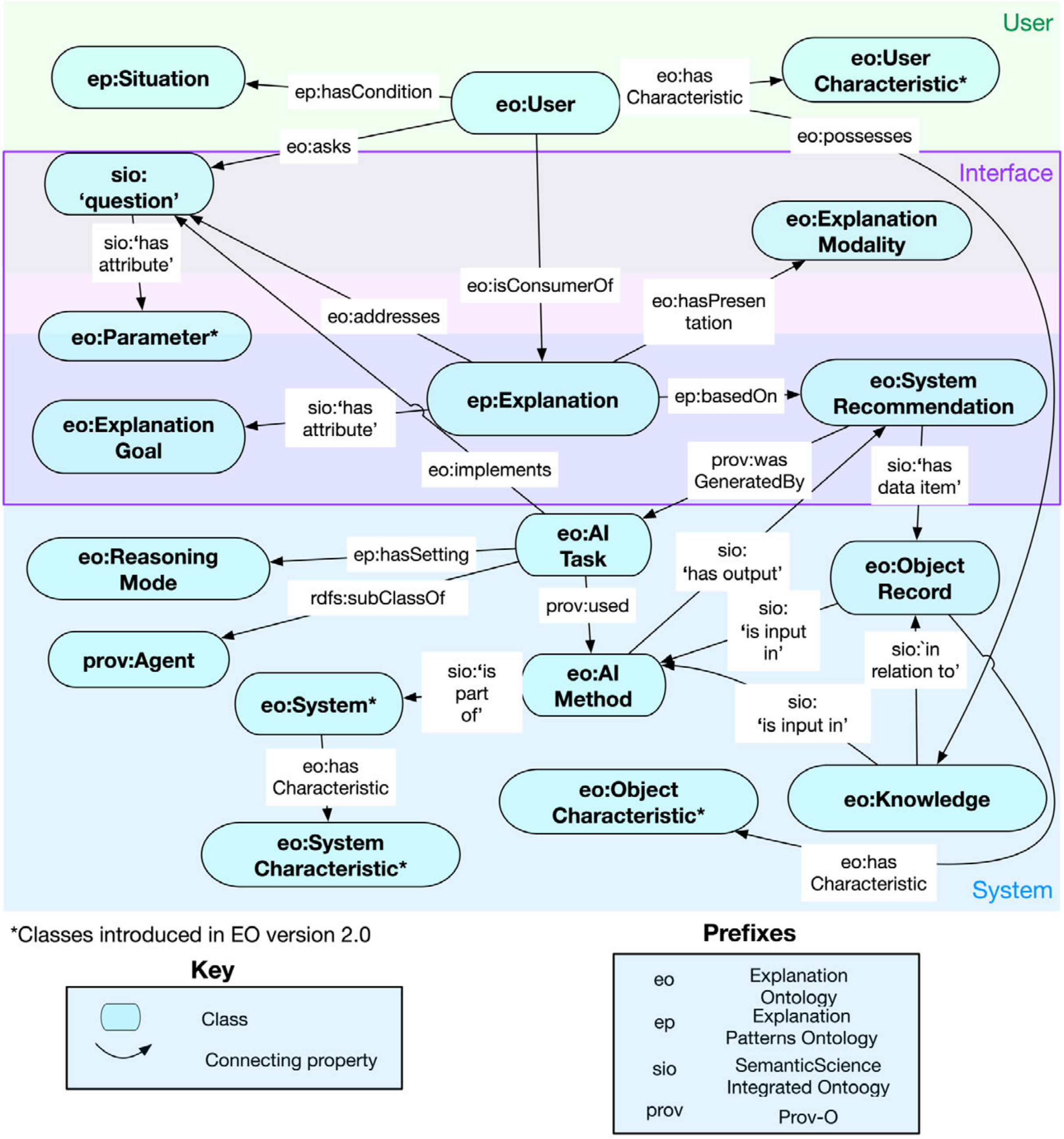


Fig. 2. Explanation ontology overview with key classes separated into three overlapping attribute categories (depicted as colored rectangles).

System Attributes: System attributes encapsulate the concepts surrounding the AI 'system' used to generate recommendations and produce explanations. 'Explanation's are based on 'system recommendation's, which in turn are generated by some 'AI task.' 'AI task's are analogous to the high-level operations that an AI system may perform (e.g., running a reasoning task to generate inferences or running an explanation task to generate explanations about a result). The 'AI task' relates to user attributes as it addresses the 'question' that a user is asking. Another important class we capture as part of system attributes, is that of the 'AI method', that 'AI tasks' use, to produce the 'system recommendation' that an 'explanation' is dependent on. Further, from a data modeling perspective, the ‘object record’ class helps us include contributing objects in the ‘system recommendation’ that could further have ‘knowledge’ and ‘object characteristic’ of their own, contributing to explanations. For example, in a patient’s risk prediction (an instance of ‘system recommendation’), the patient is an object record linked to the prediction, and an explanation about the patient’s risk refers to both the prediction and the patient. Also, within the system attributes, various additional concepts such as ‘reasoning mode’ and ‘system characteristic’ are modeled to capture further details about the AI system’s operations and how they relate to each other. Maintaining such system provenance helps system designers debug explanations and their dependencies.

Interface Attributes: Lastly, interface attributes capture an intersection between user and system attributes that can be directly interacted with on a user interface (UI). From an input perspective, these attributes consist of the ‘question’ that the user asks, the ‘explanation goal’ they want to be fulfilled by the explanation, and the ‘explanation modality’ they prefer. From the content display perspective, we capture the system attributes that might be displayed on the UI, such as the ‘explanation’ itself and the ‘system recommendation’ it is based on.

This mid-level model of the explanation space can be further extended by adding sub-class nodes to introduce more specific extensions for the entities where they exist. For example, both the ‘Knowledge’ and ‘AI Method’ classes have several sub-classes to capture the different types of ‘knowledge’ and ‘AI methods’ that explanations can be dependent on and are generated by, respectively. An example of the ‘AI Method’ hierarchy and its interactions to support the generation of system recommendations upon which explanations are based can be seen in Fig. 3.

EO Modeling Summary and Intended Usage: In essence, modeling the explanation space, as we have in the EO, can primarily serve two purposes. One to helps us, the EO developers, represent equivalent class restrictions for different explanation types which would allowing explanations to be classified into any of these types (Section 3.2). We introduce the supported explanation types within the EO and their equivalent class restrictions shortly in Section 3.2. A second purpose of the EO modeling is to help system designers instantiate EO’s classes to compose user-centered explanations whose provenance can be traced back to multiple dependencies spanning system, interface, and user spaces. For a system designer who wants to use the EO model to represent explanations in their domain-specific knowledge graphs (KG), they would only need to include an import statement for the EO in their KG (<owl:imports rdf:resource=”https://purl.org/heals/eo/2.0.0”/>). Upon representing explanations using the EO model (as seen in Fig. 2 and Fig. 3) and guidance which is presented in Section 5, they should be able to run a reasoner over their KG to infer explanations into one of our explanation types. A system designer could also root their domain-specific concepts in the EO’s high-level concepts, e.g., for a food use-case, an ingredient could be defined as a subclass of object characteristic, and in this way these concepts could also be considered for explanations. More examples of these domain specific instantiations can be viewed in Section 4.

Design Choices: For the design of the EO, we followed the principle of ontology reuse by only introducing classes and properties where they didn’t exist already. However, we used a policy of careful reuse in that we introduced classes if they were not a part of well-used and accepted scientific ontologies, mainly the OBO Foundry ontologies [52] or explainable AI-specific ontologies such as the explanations pattern (EP) ontology [58]. We found that we had to borrow from several ontologies, including The National Cancer Thesaurus Institute Ontology (NCIT) [7] and the Computer Science Ontology (CSO) [51], to support the attributes that contribute to explanations. From a first-principles reuse perspective, and because the EO is an ontology in the technical space, we build our ontology upon and import widely-used and standard science ontologies, including the SemanticScience Integrated Ontology (SIO) [22] and the Provenance Ontology (Prov-O). For the other ontologies that we reuse classes and properties from, we use the Minimum Information to Reference a Term (MIREOT) method [15] to include only the minimum information that we need to use the class or property within the EO. We report statistics on the overall composition of the EO, including the number of classes, objects, and data properties, both from an active import closure and introduced perspectives in Table 2.

Finally, we have tested the Pellet reasoner on the EO and our ontology can be browsed using the Protege 5.5 desktop tool [49].

3.2. *Explanation types*

We refer to explanation types in line with the different kinds of explanations that have been referred to in the explanation sciences literature, arising in such fields as “law, cognitive science, philosophy, and the social sci-

Table 2

Statistics on the composition of the Explanation Ontology. These counts are taken by loading the EO into the ontology editing tool, Protege [49] and choosing the active closure of the ontology with imported ontologies view option [50]. We calculated the counts of classes and properties we introduced by commenting out the import statements in the ontology file. The introduced classes and properties are indicated by a * in the table

Metrics	Count
Classes	1707
Classes Introduced*	135
Object Properties	283
Object Properties Introduced*	52
Data Properties	8
Data Properties Introduced*	2
Equivalent Class Axioms	85
Equivalent Class Axioms Introduced*	35
Instances	17
Instances Introduced*	13

ences" [41]. These include such things as contrastive, scientific, and counterfactual explanations. We found that these explanations are well defined in adjacent fields of the explanation sciences and more rarely in computer science. We performed a literature review looking for explanations that serve different purposes, address different questions, and are populated by different components (such as 'cases' for case-based explanations and evidence for 'scientific explanations') in the hope of refining their definitions to make these explanation types easier to generate by computational means. We previously released these explanation types and their definitions as a taxonomy [11], and we now encode these explanation types, definitions, and sufficiency conditions in the EO. The explanation types we support in the EO, their definitions, and sufficiency conditions can be browsed in Table 3, Table 4, Table 5 and Table 6.

In the EO, against each explanation type, we encode the sufficiency conditions as equivalent class axioms, allowing instances that fit these patterns to be automatically inferred as instances of the explanation types. An example of this can be seen in Listing 1, where we express the equivalent class axiom for a 'contextual explanation' using the core EO model and its high-level classes, such as 'system recommendation' and `'object record'`. We find that defining the equivalent class restrictions using top-level classes allows the subsumption of sub-classes of these top-level classes into the explanation type restrictions as well. Patterns of such subsumptions can be seen in our use case descriptions (Section 4). Further, if system designers want to familiarize themselves with the components that are necessary for each explanation type, we suggest that they browse the sufficiency conditions, unless they have familiarity with the OWL ontology language. The equivalent class restrictions are a logical translation of the sufficiency conditions, so the modeling of explanations based on an understanding of the sufficiency conditions should prove sufficient. Against some explanation types, we also maintain what 'AI methods' can generate them or what types of 'knowledge' they are dependent upon, so if a system designer were looking to support certain explanation types in their systems, they can plan ahead to include these methods and/or knowledge types.

Also, in addition to the *nine* different explanation types that we previously supported in the EO (Table 3 and Table 4), we have added *six* new explanation types mentioned in Zhou et al.'s recent paper [64] (Table 5 and Table 6). The addition of these explanation types also prompted the support for explanation methods as subclasses of 'AI Method.' Over the past decade, with the focus on trustworthy AI, there have been several developments in explanation methods or model explainers [4,38,48,61,62]. Part of our goal for the expansion of the EO was to be of use in explainability toolkits, so we reviewed a comprehensive set of explanation methods that are a part of the AI Explainability 360 toolkit [32]. We encode the outputs of these explanation methods, i.e., model explanations and their subtypes (e.g., local, global, static, and interactive explanations) as subclasses of 'system recommendation.' The modeling of model explanations and explanation methods can be seen in Fig. 3. We also capture the dependencies on user-centered explanations we define in Table 3 on these 'model explanations.' Further, we observe that the

Table 3

An overview of 5/9 previously supported explanation types, their simplified descriptions, example questions they can address (in bold, within the description column), and their sufficiency conditions expressed in natural language

Explanation Type	Description	Sufficiency Conditions
Case Based	Provides solutions that are based on actual prior cases that can be presented to the user to provide compelling support for the system's conclusions, and may involve analogical reasoning, relying on similarities between features of the case and of the current situation. **"To what other situations has this recommendation been applied?"**	Is there at least one other prior case (*'object record'*) similar to this situation that had an *'explanation'*? Is there a similarity between this case, and that other case?
Contextual	Refers to information about items other than the explicit inputs and output, such as information about the user, situation, and broader environment that affected the computation. **"What broader information about the current situation prompted the suggestion of this recommendation?"**	Are there any other extra inputs that are not contained in the *'situation'* description itself? And by including those, can better insights be included in the *'explanation'*?
Contrastive	Answers the question "Why this output instead of that output," making a contrast between the given output and the facts that led to it (inputs and other considerations), and an alternate output of interest and the foil (facts that would have led to it). **"Why choose option A over option B that I typically choose?"**	Is there a *'system recommendation'* that was made (let's call it A)? What facts led to it? Is there another *'system recommendation'* that could have happened or did occur, (let's call it B)? What was the *'foil'* that led to B? Can A and B be compared?
Counterfactual	Addresses the question of what solutions would have been obtained with a different set of inputs than those used. **"What if input A was over 1000?"**	Is there a different set of inputs that can be considered? If so what is the alternate *'system recommendation'*?
Everyday	Uses accounts of the real world that appeal to the user, given their general understanding and worldly and expert knowledge. **"Why does option A make sense"**	Can accounts of the real world be simplified to appeal to the user based on their general understanding and *'knowledge'*?

user-centered explanation types can depend on more than one model explanation and users need more context and knowledge to consume these model explanations.

4. Use cases

To demonstrate the utility of the EO as a general-purpose ontology to represent explanations, we show how the EO's model can be used to compose explanations in five different use cases spanning food, healthcare, and finance domains (Table 7). All of these use cases involve data available in the open domain. The first use case is in the food domain and based off of the FoodKG[5] [30], and the rest of the use cases are from among those listed on the AIX-360 website[6] [32]. In AIX-360 usecases, a suite of 'explanation methods' belonging to the AIX-360 toolkit [2,3] are run. Additionally, each of the AIX-360 use cases has a technical description or Jupyter notebook tutorial that system designers can comprehend and utilize to build the instance KGs.

In each of our five use cases, we assume that AI methods, primarily ML methods, have already been run and generated 'system recommendations'. The output of the AI methods may include enough data or may need to combined with content from a background KG (such as in the food use case) to generate the different explanation types that we support in the EO. Each of the use case has a set of example questions for which different explanation methods and/or ML methods are run. A listing of these example questions, as well as the explanation types inferred

[5] https://foodkg.github.io

[6] https://aix360.mybluemix.net

Table 4

An overview of 4/9 previously supported explanation types, their simplified descriptions, example questions they can address (in bold, within the description column), and their sufficiency conditions expressed in natural language

Explanation Type	Description	Sufficiency Conditions
Scientific	References the results of rigorous scientific methods, observations, and measurements. **"What studies have backed this recommendation?"**	Are there results of rigorous *'scientific methods'* to explain the situation? Is there *'evidence'* from the literature to explain this *'system recommendation'*, *'situation'* or *'object record'*?
Simulation Based	Uses an imagined or implemented imitation of a system or process and the results that emerge from similar inputs. **"What would happen if this recommendation is followed?"**	Is there an *'implemented'* imitation of the *'situation'* at hand? Does that other scenario have inputs similar to the current *'situation'*?
Statistical	Presents an account of the outcome based on data about the occurrence of events under specified (e.g., experimental) conditions. Statistical explanations refer to numerical evidence on the likelihood of factors or processes influencing the result. **"What percentage of people with this condition have recovered?"**	Is there *'numerical evidence'*/likelihood account of the *'system recommendation'* based on data about the occurrence of the outcome described in the recommendation?
Trace Based	Provides the underlying sequence of steps used by the system to arrive at a specific result, containing the line of reasoning per case and addressing the question of why and how the application did something. **"What steps were taken by the system to generate this recommendation?"**	Is there a record of the underlying sequence of steps (*'system trace'*) used by the *'system'* to arrive at a specific *'recommendation'*?

from running the reasoner on each use case's knowledge graph, is provided in Table 7. Here, we describe each of the five use cases in a depth sufficient enough so that a system designer who wants to use the EO to instantiate outputs from their own use cases can seek guidance on building their use case KGs from the patterns that we use for these five exemplar KGs. Our use case KG files can also be downloaded from our resource website (Section 5.1).

4.1. Food recommendation

In the food recommendation use case, aimed at recommending foods that fit a person's preferences, dietary constraints, and health goals, we have previously published a customized version of the EO specifically for the food domain, the Food Explanation Ontology [45]. With the updates to the EO, we are now able to support the modeling of contextual and contrastive examples natively in the EO, whose capabilities were previously only in FEO as depicted in [45]. In the food use case, a knowledge base question-answering (QA) system [14] has been run and outputs answers to questions like "What should I eat if I am on a keto diet?" However, a standard QA system cannot directly address more complex questions that require a reasoner to be run on the underlying knowledge graph to generate inferred content. For example, questions such as whether or not one can eat a particular recipe, like "spiced cauliflower soup", might not be easily addressable by the QA system because it doesn't know to specifically look for inferred information such as the seasonal context and availability of ingredients. The EO becomes useful here because the restrictions defined against the 'environmental context' class can classify any seasonal characteristics defined against food as environmental context concerning that food 'object record.' Hence, when we define an explanation for "Why one should eat spiced cauliflower soup" to be based on a seasonal characteristic, our EO reasoner can then classify the season instance to be an 'environmental context' and therefore classify the explanation to be a 'contextual explanation.' The contextual explanation instance and its dependencies can be viewed in Fig. 4. Similarly, to address another question, "Why is creamed broccoli soup recommended over tomato soup", we extract 'facts' supporting creamed broccoli soup and 'foils' or 'facts' not in support of tomato soup. Then from this representation, our EO reasoner can infer that an 'explanation' depending on the 'system recommendation' and that encapsulates reasons in support of creamed broccoli soup over tomato soup, to be a contrastive explanation. More broadly, if a system designer can define explanations dependent on food system recommendations and link

Table 5

An overview of 3/6 new explanation types described in Zhou et al. [64] that we encode in the Explanation Ontology version 2.0

Explanation Type	Description	Sufficiency Conditions
Data	Focuses on what the data is and how it has been used in a particular decision, as well as what data and how it has been used to train and test the ML model. This type of explanation can help users understand the influence of data on decisions. **"What the data is?", "How it has been used in a particular decision?", "How has the data been used to train the ML model?"**	Is there a 'system recommendation' from an 'AI method' that has as input, a 'dataset' or part of it? Is there a 'system recommendation' that includes 'object records' that are used to train / test the 'AI method'?
Rationale	About the "why" of an ML decision and provides reasons that led to a decision, and is delivered in an accessible and understandable way, especially for lay users. If the ML decision was not what users expected, rationale explanations allows users to assess whether they believe the reasoning of the decision is flawed. While, if so, the explanation supports them to formulate reasonable arguments for why they think this is the case. **"Why was this ML decision made and provide reasons that led to a decision?"**	Is there a 'system recommendation' from an 'AI method' that has a 'system trace'? Is there a 'local explanation' output that an 'explanation' is based on?
Safety and Performance	Deals with steps taken across the design and implementation of an ML system to maximise the accuracy, reliability, security, and robustness of its decisions and behaviours. Safety and performance explanations help to assure individuals that an ML system is safe and reliable by explanation to test and monitor the accuracy, reliability, security, and robustness of the ML model. **"What steps were taken to ensure robustness and reliability of system?", "How has the data been used to train the ML model?", "What steps were taken to ensure robustness and reliability of AI method?", "What were the plans for the system development?"**	Is there a 'system recommendation' from an 'AI method' that is part of a 'system' that exposes its design 'plans'? Is there a 'system recommendation' that includes 'object records' that are used to train / test the 'AI method'?

the characteristics related to an 'object record' contained in the recommendations. Then, when a reasoner is run on the EO, it could look for patterns that can match the user-centered explanation types we support and populate the explanation types whose patterns match the KG content.

4.2. *Proactive retention*

In the proactive retention use case,[7] the objective is to learn the rules for employee retention that could signal to an employing organization whether or not employees are likely to have retention potential. Since these rules involve a deep understanding of the contributing attributes of the employee dataset, a domain expert is best adept at providing these rules. Potentially, a supervised ML method could be run to learn the rules for other unlabeled instances. The AIX-360 toolkit supports a TED Cartesian Explainer algorithm [31] that can learn from rules that are defined against a few cases and predict the rules for others. More specifically, in the proactive retention use case KG, the TED Cartesian Explainer method is defined as an instance of 'Providing rationale method' in the EO by virtue of the definition for this 'explanation method' subclass, and that the TED Explainer provides local explanations for each instance. Additionally, since the TED Explainer provides 'system recommendations' for every employee retention output pair, we save this dependency of explanations on local instance-level system recommendations in the KG and define explanations based on these retention rule explanations. As can be seen from the 'explanation types supported' column of the proactive retention row, a reasoner infers rationale explanations, see Fig. 5, from

[7] Proactive retention use case: https://nbviewer.org/github/IBM/AIX360/blob/master/examples/tutorials/retention.ipynb.

Table 6

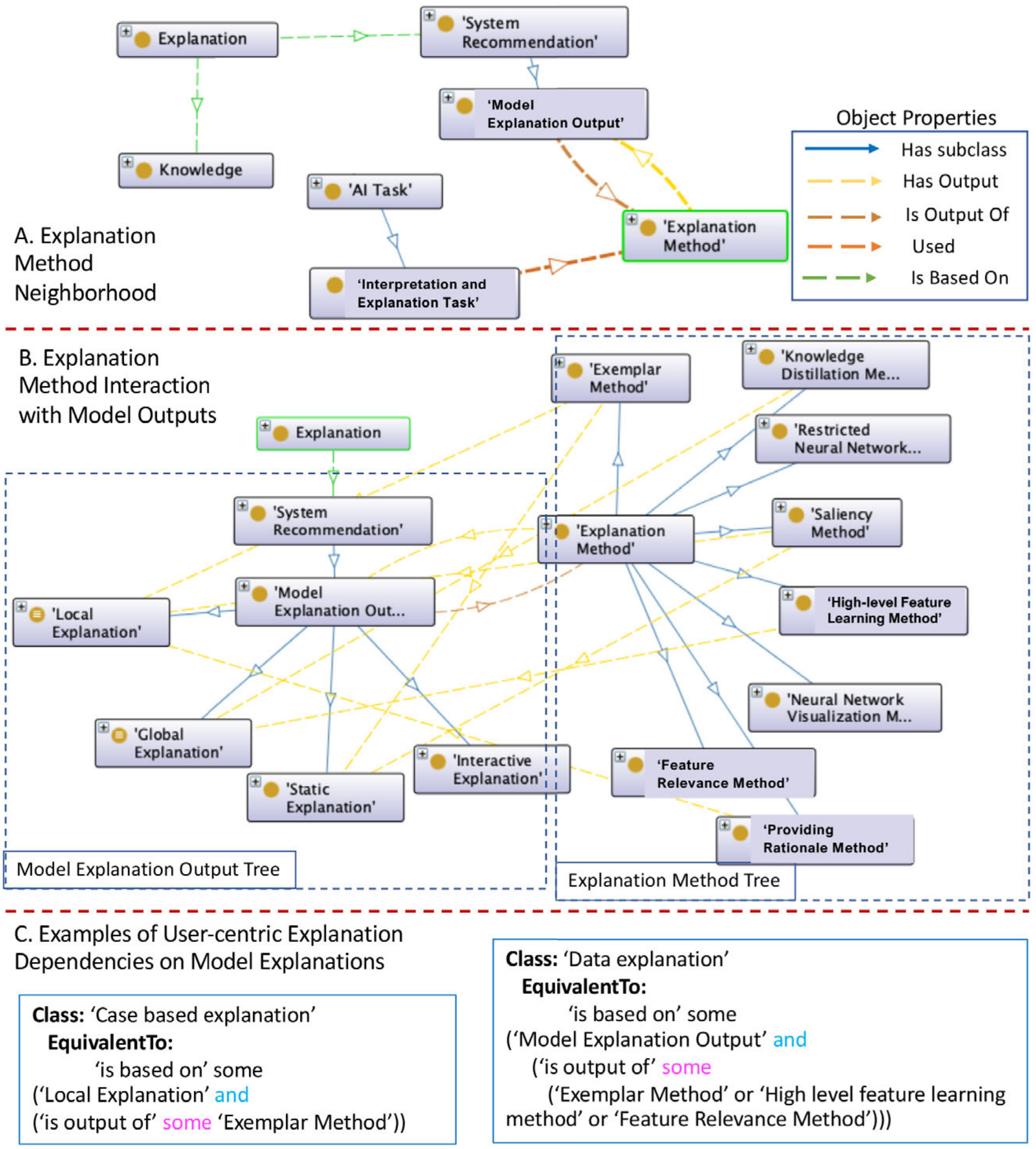
An overview of 3/6 new explanation types described in Zhou et al. [64] that we encode in the Explanation Ontology version 2.0

Explanation Type	Description	Sufficiency Conditions
Impact	Concerns the impact that the use of a system and its decisions has or may have on an individual and on a wider society. Impact explanations give individuals some power and control over their involvement in ML-assisted decisions. By understanding the possible consequences of the decision, an individual can better assess their participation in the process and how the outcomes of the decision may affect them. **“What is the impact of a system recommendation?”, “How will the recommendation affect me?”**	Is there a ‘system recommendation’ from an ‘AI method’ that has a ‘statement of consequence’?
Fairness	Provides steps taken across the design and implementation of an ML system to ensure that the decisions it assists are generally unbiased, and whether or not an individual has been treated equitably. Fairness explanations are key to increasing individuals’ confidence in an AI system. It can foster meaningful trust by explaining to an individual how bias and discrimination in decisions are avoided. **“Is there a bias consequence of this system recommendation?”, “What data was used to arrive at this decision?”**	Is there a ‘system recommendation’ from an ‘AI method’ that has a ‘statement of consequence’? Is there a ‘dataset’ in the ‘system recommendation’ the explanation is based on?
Responsibility	Concerns “who” is involved in the development, management, and implementation of an ML system, and “who” to contact for a human review of a decision. Responsibility explanations help by directing the individual to the person or team responsible for a decision. It also makes accountability traceable. **“Who is involved in the development, management, and implementation of an ML system?”, “Who to contact for a human review of a decision?”**	Is there a ‘system recommendation’ from an ‘AI method’ that was part of a ‘system’, whose ‘system developer’ is known?

```
Class: eo:ContextualExplanation
 EquivalentTo:
  ((ep:‘is based on’ some
  (eo:‘Contextual Knowledge’
   and ((sio:‘is attribute of’ some Situation) or (sio:‘in relation to’ some Situation)))) or
   (ep:‘is based on’ some
  (eo:‘Contextual Knowledge’
   and ((sio:‘is attribute of’ some eo:‘Object Record’) or (sio:‘in relation to’ some eo:‘Object
         Record’)))) or
   (ep:‘is based on’ some
  (eo:‘Contextual Knowledge’
   and (sio:‘has attribute’ some eo:‘Object Record’))))
 and (ep:‘is based on’ some
  (eo:‘System Recommendation’ or eo:‘Object Record’))
 SubClassOf:
  ep:Explanation

Class: ep:Explanation
  SubClassOf:
    sio:‘computational entity’,
    ep:isBasedOn some eo:Knowledge,
    ep:isBasedOn some eo:SystemRecommendation,
    ep:isConceptualizedBy some eo:AITask
```

Listing 1. OWL expression of the representation of a *‘contextual explanation’* in Manchester syntax


Explanation
'System Recommendation'
'Model Explanation Output'
Knowledge
'AI Task'
'Explanation Method'
'Interpretation and Explanation Task'
A. Explanation Method Neighborhood
Object Properties
Has subclass
Has Output
Is Output Of
Used
Is Based On
B. Explanation Method Interaction with Model Outputs
Explanation
'System Recommendation'
'Model Explanation Out...
'Local Explanation'
'Global Explanation'
'Static Explanation'
'Interactive Explanation'
Model Explanation Output Tree
'Exemplar Method'
'Knowledge Distillation Me...
'Restricted Neural Network...
'Explanation Method'
'Saliency Method'
'High-level Feature Learning Method'
'Neural Network Visualization M...
'Feature Relevance Method'
'Providing Rationale Method'
Explanation Method Tree
C. Examples of User-centric Explanation Dependencies on Model Explanations

Class: 'Case based explanation'
EquivalentTo:
'is based on' some
('Local Explanation' and
('is output of' some 'Exemplar Method'))

Class: 'Data explanation'
EquivalentTo:
'is based on' some
('Model Explanation Output' and
('is output of' some
('Exemplar Method' or 'High level feature learning method' or 'Feature Relevance Method')))

Fig. 3. Ontograf [44] visualization of the explanation methods we support in the explanation ontology (Part A and B). The ontology includes terms that can be used to map the outputs of these methods to support population of user-centered explanation types (Part B and C).

the proactive retention KG. These rationale explanations are inferred that way because they match the equivalent class restriction of the 'rationale explanation' class in the EO, wherein we look for rationales or traces supporting a system recommendation, which in this case are the rules providing rationales for the employee retention output.

Table 7

A listing of example questions and inferred explanation types supported by each use case

Use case	Example Questions	Explanation Types Inferred
Food Recommendation	Why should I eat spiced cauliflower soup? Why creamed broccoli soup over tomato soup?	Contextual and Contrastive
Proactive Retention	What is the retention action outcome for this employee?	Rationale
Health Survey Analysis	Who are the most representative patients in this questionnaire? Which questionnaires have the highest number of most representative patients?	Case based and Contextual
Medical Expenditure	What are the rules for expenditure prediction? What are patterns for high-cost patients?	Data
Credit Approval	What are the rules for credit approval? What are some representative customers for credit? What factors if present and if absent contribute most to credit approval?	Data, Case based and Contrastive

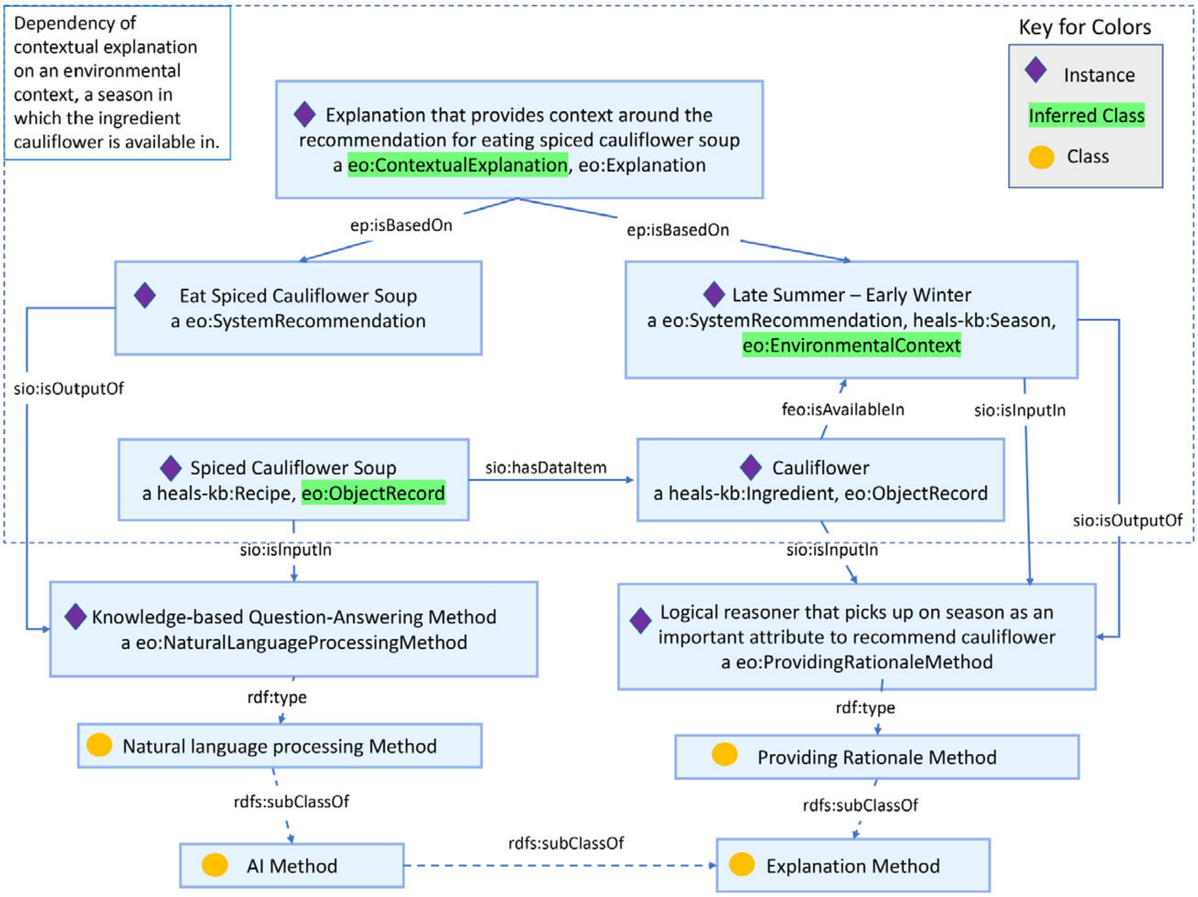

Dependency of contextual explanation on an environmental context, a season in which the ingredient cauliflower is available in.
Key for Colors
Instance
Inferred Class
Class
Explanation that provides context around the recommendation for eating spiced cauliflower soup a eo:ContextualExplanation, eo:Explanation
ep:isBasedOn
ep:isBasedOn
Eat Spiced Cauliflower Soup a eo:SystemRecommendation
Late Summer – Early Winter a eo:SystemRecommendation, heals-kb:Season, eo:EnvironmentalContext
sio:isOutputOf
feo:isAvailableIn
sio:isInputIn
Spiced Cauliflower Soup a heals-kb:Recipe, eo:ObjectRecord
sio:hasDataItem
Cauliflower a heals-kb:Ingredient, eo:ObjectRecord
sio:isOutputOf
sio:isInputIn
sio:isInputIn
Knowledge-based Question-Answering Method a eo:NaturalLanguageProcessingMethod
Logical reasoner that picks up on season as an important attribute to recommend cauliflower a eo:ProvidingRationaleMethod
rdf:type
rdf:type
Natural language processing Method
Providing Rationale Method
rdfs:subClassOf
rdfs:subClassOf
AI Method
rdfs:subClassOf
Explanation Method

Fig. 4. Annotated snippet of a contextual explanation instance from the food recommendation use case knowledge graph. Ontology prefixes used in the figure are presented in Table 1 and upper-level classes used from the Explanation Ontology model are introduced in Fig. 2 and Fig. 3.

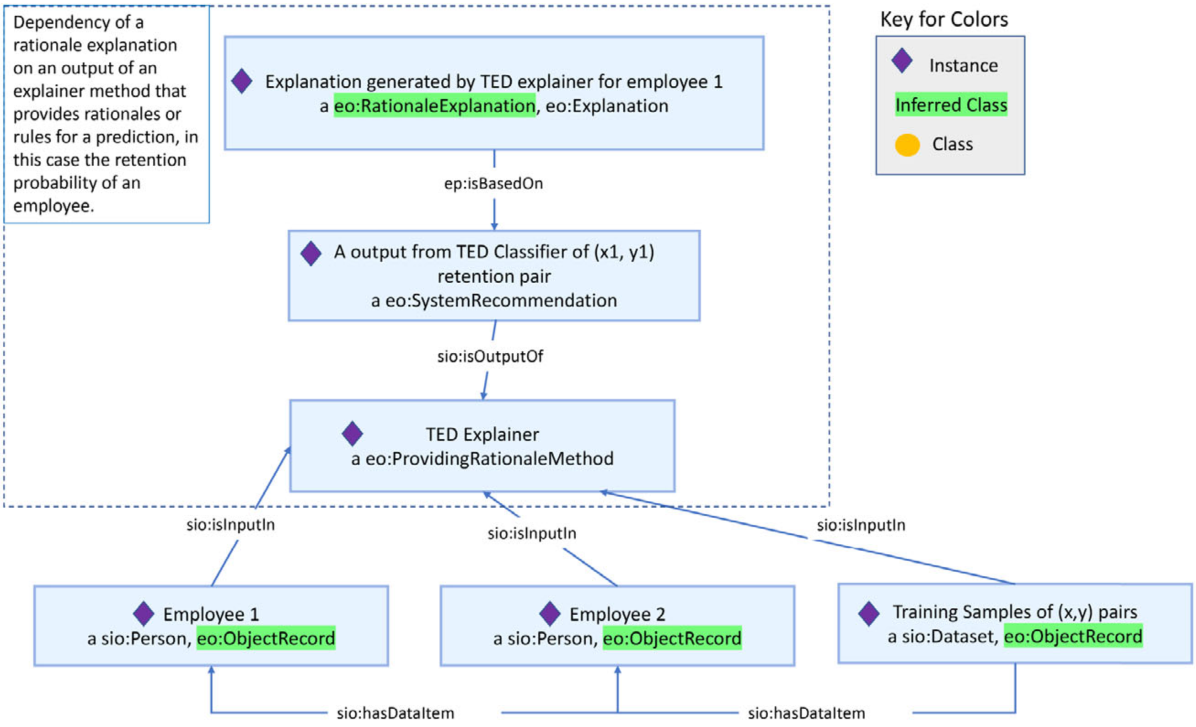

Dependency of a rationale explanation on an output of an explainer method that provides rationales or rules for a prediction, in this case the retention probability of an employee.
Key for Colors
Instance
Inferred Class
Class
Explanation generated by TED explainer for employee 1
a eo:RationaleExplanation, eo:Explanation
ep:isBasedOn
A output from TED Classifier of (x1, y1) retention pair
a eo:SystemRecommendation
sio:isOutputOf
TED Explainer
a eo:ProvidingRationaleMethod
sio:isInputIn
sio:isInputIn
sio:isInputIn
Employee 1
a sio:Person, eo:ObjectRecord
Employee 2
a sio:Person, eo:ObjectRecord
Training Samples of (x,y) pairs
a sio:Dataset, eo:ObjectRecord
sio:hasDataItem
sio:hasDataItem

Fig. 5. Annotated snippet of a rationale explanation instance from the proactive retention use case knowledge graph. Ontology prefixes used in the figure are presented in Table 1 and upper-level classes used from the Explanation Ontology model are introduced in Fig. 2 and Fig. 3.

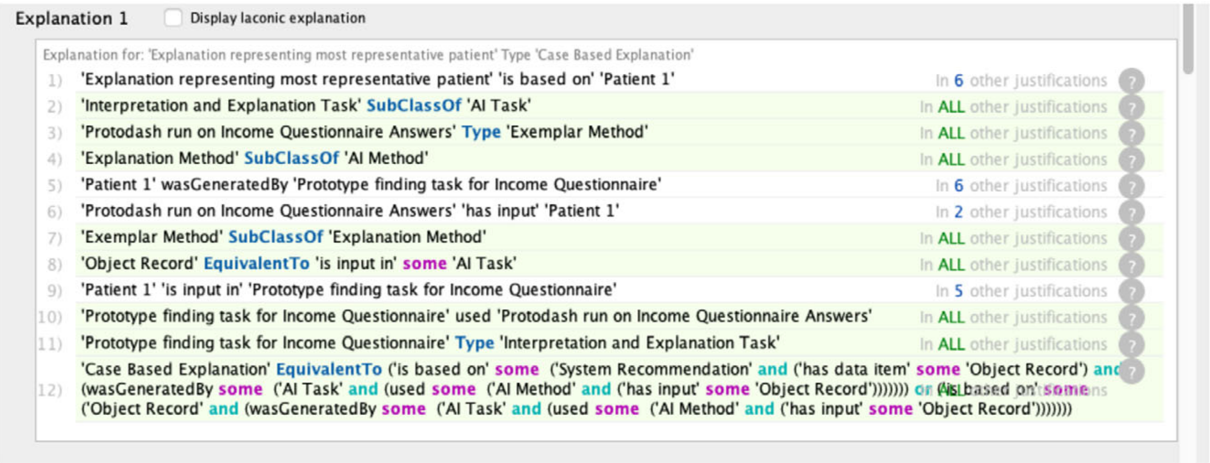
4.3. *Health survey analysis*

The health survey analysis use case[8] utilizes the National Health and Nutritional Exam Survey (NHANES) dataset [8]. The objective in this use case entails two ‘explanation tasks’: (1) find the most representative patients for the income questionnaire, and (2) find the responses that are most indicative/representative of the income questionnaire. The Protodash method [29], an ‘Exemplar explanation method,’ that finds representative examples from datasets, is run for both these tasks. However, in the second task, an additional data interpretation or summarization method is run to evaluate the prototype patient cases of questionnaires for how well they correlate to responses in the income questionnaire. Hence, when we instantiate the outputs of these two tasks, we use different chains of representations to indicate the dependencies of the explanations of the two questions that are addressed by these tasks. More specifically, this chaining would mean that we define that the ‘system recommendation’ of the ‘summarization’ method instance to be dependent on or use as input the ‘system recommendations’ of ‘Protodash’ instances.

Additionally, as can be seen from the ‘explanation types’ supported column of Table 7 against the health survey analysis, there are two explanation types inferred upon running a reasoner against the NHANES KG: case-based and contextual. The explanation finding for the most representative patients of the income questionnaire is case-based since it contains patient case records. Further, the trail of outputs that contributed to the explanation to be classified as a case-based explanation can be seen in Fig. 6.

Still, the classification of the explanations of the most representative questionnaire as a contextual explanation is less obvious. However, upon closer investigation of the equivalent condition defined against the ‘environmental context‘class in the EO from Listing 2, we can see how the patients’ questionnaires, a type of ‘file,’ are inferred to be the ‘environmental context’ for the patient, an ‘object record’, since the patients participate in those question-

[8] NHANES use case documentation: https://nbviewer.org/github/IBM/AIX360/blob/master/examples/tutorials/CDC.ipynb.



Explanation 1 □ Display laconic explanation

Explanation for: 'Explanation representing most representative patient' Type 'Case Based Explanation'

1) 'Explanation representing most representative patient' 'is based on' 'Patient 1' — In 6 other justifications
2) 'Interpretation and Explanation Task' SubClassOf 'AI Task' — In ALL other justifications
3) 'Protodash run on Income Questionnaire Answers' Type 'Exemplar Method' — In ALL other justifications
4) 'Explanation Method' SubClassOf 'AI Method' — In ALL other justifications
5) 'Patient 1' wasGeneratedBy 'Prototype finding task for Income Questionnaire' — In 6 other justifications
6) 'Protodash run on Income Questionnaire Answers' 'has input' 'Patient 1' — In 2 other justifications
7) 'Exemplar Method' SubClassOf 'Explanation Method' — In ALL other justifications
8) 'Object Record' EquivalentTo 'is input in' some 'AI Task' — In ALL other justifications
9) 'Patient 1' 'is input in' 'Prototype finding task for Income Questionnaire' — In 5 other justifications
10) 'Prototype finding task for Income Questionnaire' used 'Protodash run on Income Questionnaire Answers' — In ALL other justifications
11) 'Prototype finding task for Income Questionnaire' Type 'Interpretation and Explanation Task' — In ALL other justifications
12) 'Case Based Explanation' EquivalentTo ('is based on' some ('System Recommendation' and ('has data item' some 'Object Record') and (wasGeneratedBy some ('AI Task' and (used some ('AI Method' and ('has input' some 'Object Record')))))) or ('is based on' some ('Object Record' and (wasGeneratedBy some ('AI Task' and (used some ('AI Method' and ('has input' some 'Object Record')))))) — In ALL other justifications

Fig. 6. A screenshot from the Protege ontology editing tool identifying the rules that were satisfied for a particular explanation to be classified as a case based explanation.

```
Class: eo:ContextualKnowledge
 EquivalentTo:
   (‘social entity’ or sio:representation or sio:media or sio:Location)
and (eo:’has attribute’ some eo:’Object Record’)
 SubClassOf:
   eo:ContextualKnowledge
```

Listing 2. OWL expression of the *‘environmental context’* and its sufficiency conditions in Manchester syntax

naires. Hence, an explanation, such as finding the most representative questionnaire that is dependent on both the questionnaire and the representative patient cases, would be classified as a contextual explanation (see Fig. 7). Such contextual explanations can help identify which parts of the larger context were impactful in the system recommendation and help system designers and developers better explain their system workings to end-users.

4.4. Medical expenditure

In the medical expenditure use case,[9] the objective is to learn the rules for the demographic and socio-economic factors that impact the medical expenditure patterns of individuals. Hence, from the description itself, we can infer that in this use case the rules are attempting to understand the patterns in the Medical Expenditure Panel Survey (MEPS) dataset or the general behavior of the prediction models being applied on the dataset, as opposed to attempting to understand why a particular decision was made. This use case involves the use of global explanation methods from the AIX-360 toolkit [2,3], including the Boolean Rule Column Generator and Linear Rule Regression (LRR) methods. The ‘explanation method’ instances, in this case, produce explanations that are dependent on ‘system recommendations,’ which rely on the entire dataset itself. Therefore, system designers should link the ‘system recommendations’ to the dataset. We achieve this association in our MEPS KG by representing the ‘dataset’ instance as an input of the LRR and BRCG methods. Finally, as can be seen from Table 7, the reasoner can only infer data explanations (refer to Fig. 8) from the MEPS KG instances, as the explanations are dependent on the rules identified for patterns in the dataset. Hence, in a use case such as this, wherein the explanations are mainly dependent on the dataset and the patterns within them, if a system designer were to appropriately link the system recommendation that the explanation is based on to the entire dataset or a component of the same (i.e., a column,

[9] Medical expenditure use case: https://nbviewer.org/github/IBM/AIX360/blob/master/examples/tutorials/MEPS.ipynb.

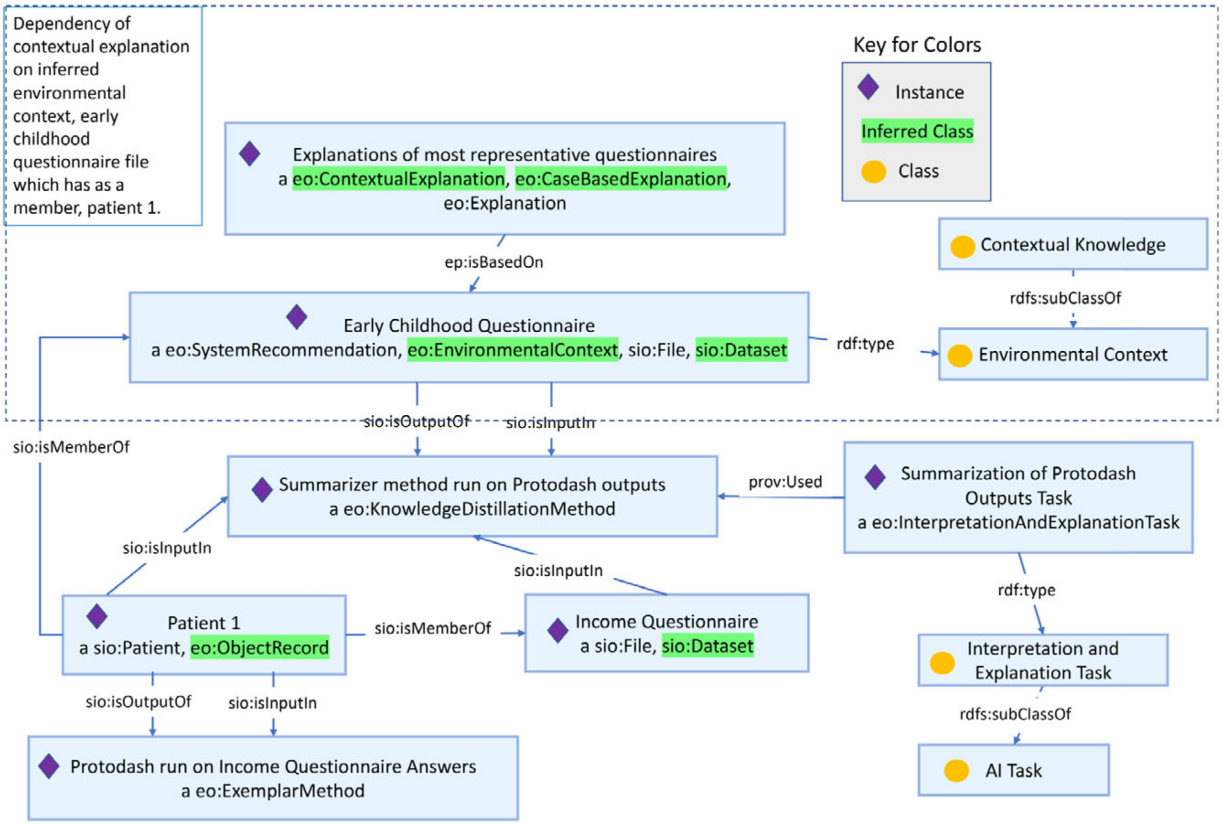

Dependency of contextual explanation on inferred environmental context, early childhood questionnaire file which has as a member, patient 1.
Key for Colors
Instance
Inferred Class
Class
Explanations of most representative questionnaires
a eo:ContextualExplanation, eo:CaseBasedExplanation, eo:Explanation
ep:isBasedOn
Contextual Knowledge
rdfs:subClassOf
Early Childhood Questionnaire
a eo:SystemRecommendation, eo:EnvironmentalContext, sio:File, sio:Dataset
rdf:type
Environmental Context
sio:isOutputOf
sio:isInputIn
sio:isMemberOf
Summarizer method run on Protodash outputs
a eo:KnowledgeDistillationMethod
prov:Used
Summarization of Protodash Outputs Task
a eo:InterpretationAndExplanationTask
sio:isInputIn
sio:isInputIn
rdf:type
Patient 1
a sio:Patient, eo:ObjectRecord
sio:isMemberOf
Income Questionnaire
a sio:File, sio:Dataset
Interpretation and Explanation Task
sio:isOutputOf
sio:isInputIn
rdfs:subClassOf
Protodash run on Income Questionnaire Answers
a eo:ExemplarMethod
AI Task

Fig. 7. Annotated snippet of a contextual explanation instance from the health survey analysis use case knowledge graph. Ontology prefixes used in the figure are presented in Table 1 and upper-level classes used from the Explanation Ontology model are introduced in Fig. 2 and Fig. 3.

row or cell), a reasoner run on the use case KG can populate data explanations of the explanation representations. Such data explanations can be helpful to understand aspects of bias and coverage in the data, which can signal to system designers and users whether their dataset is serving its intended purpose.

4.5. *Credit approval*

In the credit approval use case,[10] there are several objectives depending on the 'user' persona, including to enable data scientists to familiarize themselves with the factors that impact the credit approval outcome, for loan officers to identify prototypical cases of credit approved owners, and for customers to understand what patterns in their profile contribute the most towards their credit approval. The analyses are conducted on the FICO HELOC dataset,[11] which contains "anonymized information about Home Equity Line Of Credit (HELOC) applications made by real homeowners" [23]. We run three explanation methods: (1) BRCG and LRR to provide data scientists with the rules for credit approval ratings, (2) Protodash to provide loan officers prototypical customer cases, and (3) Contrastive Explanation Method (CEM) to provide customers with explanations to what the minimally sufficient factors in achieving good credit ('fact') are and the factors which, if changed, would change their credit ('foil'). In the medical expenditure use case, we have already shown that a system designer dealing with outputs of rule-based methods, such as BRCG and LRR, can represent the explanations dependent on 'system recommendations' generated by the methods and, particularly in this use case, define the FICO HELOC dataset as input for these methods. In the case

[10] Credit Approval use case: https://nbviewer.org/github/IBM/AIX360/blob/master/examples/tutorials/HELOC.ipynb.
[11] FICO HELOC Dataset: https://aix360.readthedocs.io/en/latest/datasets.html#id16.

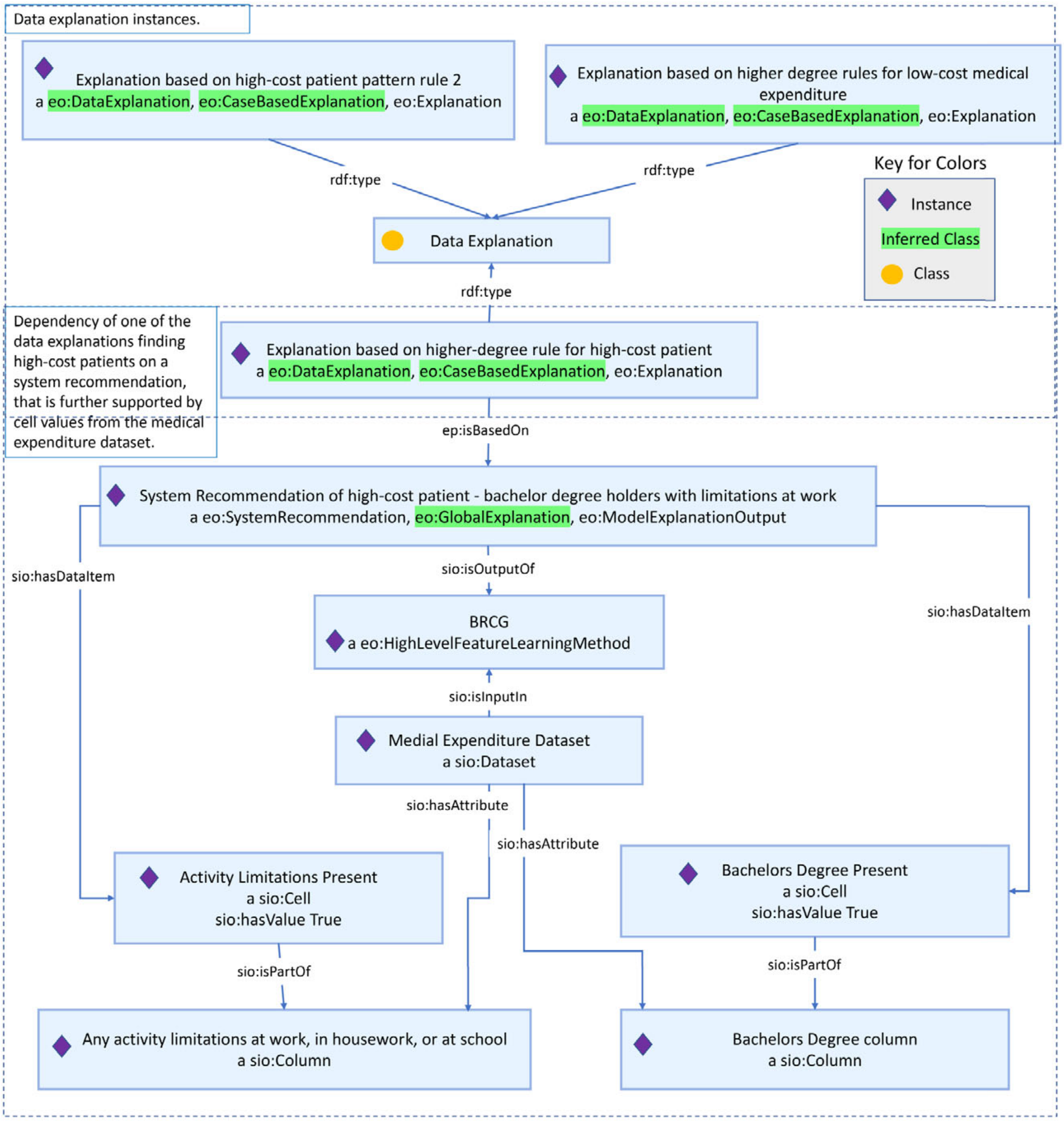

Data explanation instances.
Explanation based on high-cost patient pattern rule 2
a eo:DataExplanation, eo:CaseBasedExplanation, eo:Explanation
Explanation based on higher degree rules for low-cost medical expenditure
a eo:DataExplanation, eo:CaseBasedExplanation, eo:Explanation
rdf:type
rdf:type
Key for Colors
Instance
Inferred Class
Class
Data Explanation
rdf:type
Dependency of one of the data explanations finding high-cost patients on a system recommendation, that is further supported by cell values from the medical expenditure dataset.
Explanation based on higher-degree rule for high-cost patient
a eo:DataExplanation, eo:CaseBasedExplanation, eo:Explanation
ep:isBasedOn
System Recommendation of high-cost patient - bachelor degree holders with limitations at work
a eo:SystemRecommendation, eo:GlobalExplanation, eo:ModelExplanationOutput
sio:hasDataItem
sio:isOutputOf
sio:hasDataItem
BRCG
a eo:HighLevelFeatureLearningMethod
sio:isInputIn
Medial Expenditure Dataset
a sio:Dataset
sio:hasAttribute
sio:hasAttribute
Activity Limitations Present
a sio:Cell
sio:hasValue True
Bachelors Degree Present
a sio:Cell
sio:hasValue True
sio:isPartOf
sio:isPartOf
Any activity limitations at work, in housework, or at school
a sio:Column
Bachelors Degree column
a sio:Column

Fig. 8. Annotated snippet of a data explanation instance from the Medical Expenditure use case knowledge graph. Ontology prefixes used in the figure are presented in Table 1 and upper-level classes used from the Explanation Ontology model are introduced in Fig. 2 and Fig. 3.

of representing the identified prototypical credit approval customers, system designers can seek inspiration from the health survey analysis case and similarly represent the customer cases as instances of ‘system recommendations’ and as inputs to an ‘explanation task.’ Finally, for the outputs of CEM (see Fig. 9), we represent the factors that need to be minimally present for credit approval as ‘facts’ in support of an explanation, and the factors which, if present, flip the decision as ‘foils’ of an explanation. Such a representation would align with our definition of restrictions against the contrastive explanation class. In addition, we create *three different user instances* for the data scientist, loan officer, and customer, respectively, and further associate the questions (see credit approval row of Table 7) that

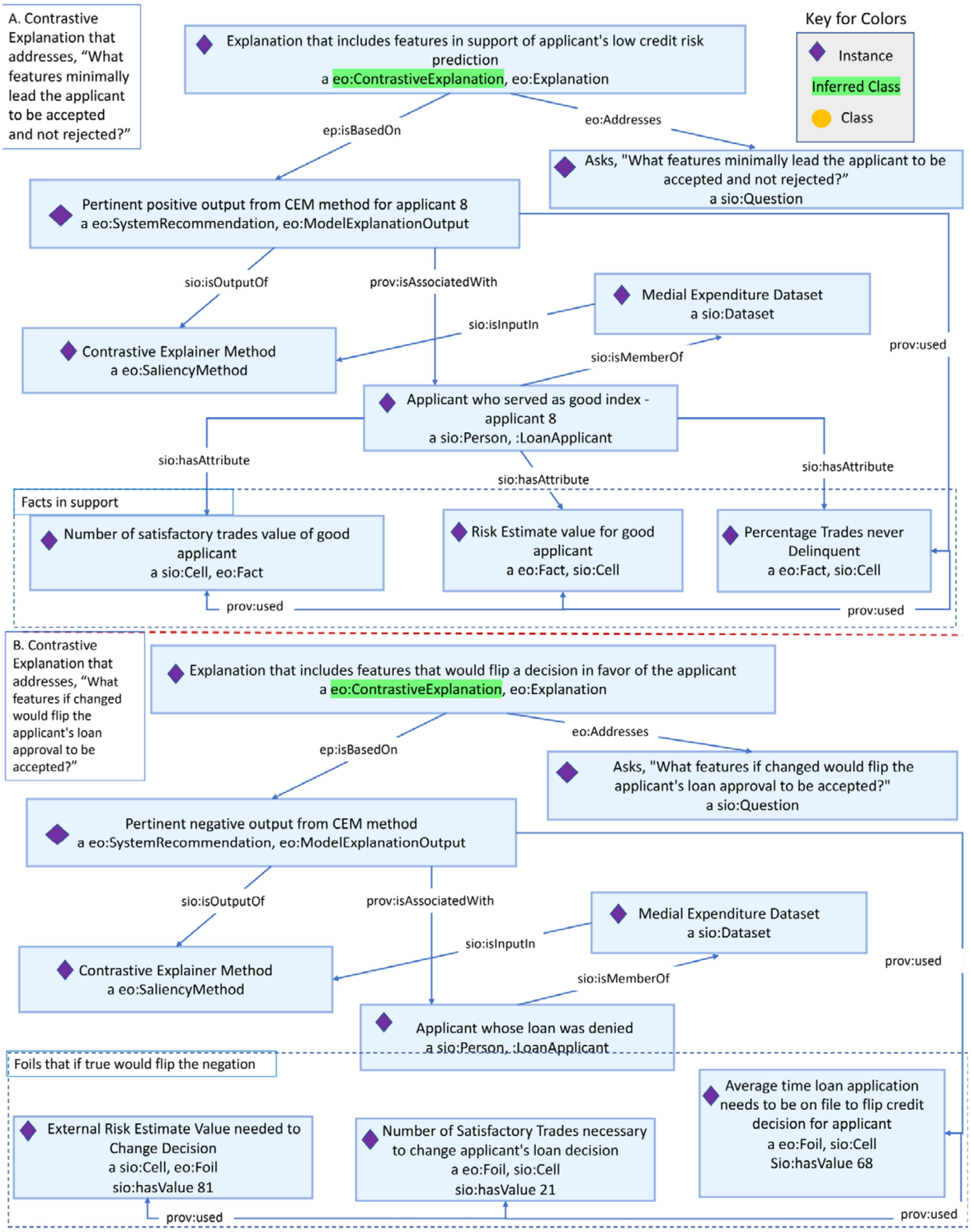

A. Contrastive Explanation that addresses, "What features minimally lead the applicant to be accepted and not rejected?"
Explanation that includes features in support of applicant's low credit risk prediction
a eo:ContrastiveExplanation, eo:Explanation
Key for Colors
Instance
Inferred Class
Class
ep:isBasedOn
eo:Addresses
Asks, "What features minimally lead the applicant to be accepted and not rejected?"
a sio:Question
Pertinent positive output from CEM method for applicant 8
a eo:SystemRecommendation, eo:ModelExplanationOutput
sio:isOutputOf
prov:isAssociatedWith
Medial Expenditure Dataset
a sio:Dataset
sio:isInputIn
prov:used
Contrastive Explainer Method
a eo:SaliencyMethod
sio:isMemberOf
Applicant who served as good index - applicant 8
a sio:Person, :LoanApplicant
sio:hasAttribute
sio:hasAttribute
sio:hasAttribute
Facts in support
Number of satisfactory trades value of good applicant
a sio:Cell, eo:Fact
Risk Estimate value for good applicant
a eo:Fact, sio:Cell
Percentage Trades never Delinquent
a eo:Fact, sio:Cell
prov:used
prov:used
B. Contrastive Explanation that addresses, "What features if changed would flip the applicant's loan approval to be accepted?"
Explanation that includes features that would flip a decision in favor of the applicant
a eo:ContrastiveExplanation, eo:Explanation
ep:isBasedOn
eo:Addresses
Asks, "What features if changed would flip the applicant's loan approval to be accepted?"
a sio:Question
Pertinent negative output from CEM method
a eo:SystemRecommendation, eo:ModelExplanationOutput
sio:isOutputOf
prov:isAssociatedWith
Medial Expenditure Dataset
a sio:Dataset
sio:isInputIn
prov:used
Contrastive Explainer Method
a eo:SaliencyMethod
sio:isMemberOf
Applicant whose loan was denied
a sio:Person, :LoanApplicant
Foils that if true would flip the negation
External Risk Estimate Value needed to Change Decision
a sio:Cell, eo:Foil
sio:hasValue 81
Number of Satisfactory Trades necessary to change applicant's loan decision
a eo:Foil, sio:Cell
sio:hasValue 21
Average time loan application needs to be on file to flip credit decision for applicant
a eo:Foil, sio:Cell
Sio:hasValue 68
prov:used
prov:used

Fig. 9. Annotated snippet of contrastive explanation instances from the credit approval use case knowledge graph. Ontology prefixes used in the figure are presented in Table 1 and upper-level classes used from the Explanation Ontology model are introduced in Fig. 2 and Fig. 3.

each of them asks via the properties supported in the EO (Fig. 2). When a reasoner is run upon the credit approval KG, we can see instances of data explanations for a data science user, case based explanations for a loan officer, and contrastive explanations for a customer.

5. Usage guidance for system designers

From the exemplar use case KGs that we have described, we aim to show how the EO can support system designers who are looking to include user-centered explanations by following the process below.

- At a high-level, given the 'system recommendations'/outputs in each of these use cases, a system designer could define them as 'system recommendation' class instances.
- Details of the specific 'AI method' and 'AI task' that generated the 'system recommendation' can be made into instances of these two classes, respectively.
- Further associations to the 'system recommendations', such as linked 'object records,' datasets, or 'characteristics' of these records, can also be represented via the EO model.
- System designers would need to define the larger 'question' addressed by the 'system recommendation(s)' upon which the 'explanations' are based. In use cases where the details of the user are present, the system designers should represent them as instances of the 'user' class and their 'user characteristics.'
- If system designers were looking to support particular user-centered explanation types, they should familiarize themselves with the sufficiency conditions for different explanation types (Table 3) in the EO. They should be able to populate the slots required for different explanation types by doing so. These equivalent restrictions can be browsed through querying our ontology using the competency questions released on our website and described in Section 6.2. After, familiarizing themselves with the EO supported explanation types, system designers should create an 'explanation' instance and link this instance to its dependencies such as the 'system recommendation' it is based on, the 'question' it addresses and in some cases to the additional 'knowledge' the explanation uses.
- Finally, when a reasoner is run on the KG, the equivalent class restrictions defined in the EO against explanation types can pick out patterns in the KG that match these restrictions and classify the system-designer-defined explanations as instances of matched explanation types. System designers can view the inferred explanation types against their explanation instances, such as the explanation snippets shown in Section 4.

Hence, to support user-centered explanations in their use cases, a system designer would often need to identify and create instances of the system outputs, their interacting attributes, and their system provenance regarding what methods generate them and the user attributes, if present, as instances of the EO classes. Additionally, supposing documentation of the use cases exists, such as in the AIX-360 use cases, system designers could fairly quickly use this documentation along with representation patterns in our exemplar use case KGs to build their use case KGs using the EO.

Currently, system designers would need to apply this guidance to build their use case KGs using an ontology editing tool like Protege. We are investigating and planning on programmatic approaches like APIs to make the creation process more automated.

5.1. *Resource contributions*

We contribute the following publicly available artifacts: our expanded Explanation Ontology with the logical formalizations of the different explanation types and SPARQL queries to evaluate the competency questions, along with the applicable documentation, all available on our resource website. On our open-source Github repository, we also release our KG files (and the inferred versions too), for the five new use cases described in this paper. These resources, listed in Table 8, are useful for anyone interested in building explanation facilities into their systems.

The ontology has been made available as an open-source artifact under the Apache 2.0 license [56] and we maintain all our artifacts on our Github repository. We also maintain a persistent URL for our ontology, hosted on the PURL service. All the relevant links are listed below in Table 8.

Table 8

Links to resources we have released and refer to in the paper

Resource	Link to Resource
Resource Website	http://tetherless-world.github.io/explanation-ontology
EO PURL URL	https://purl.org/heals/eo
Github Repository	https://github.com/tetherless-world/explanation-ontology

6. Evaluation

Our evaluation is inspired by ontology evaluation techniques proposed in Muhammad et al.'s comprehensive ontology evaluation techniques review paper [1]. They introduce an ontology evaluation taxonomy that combines evaluation techniques that each reveal different perspectives of the ontology, such as application-based, metric-based, user-based, and logic/rule-based evaluation techniques. These evaluation techniques in Muhammad *et al.'s* taxonomy are a collection of techniques proposed by four different well-cited papers, including Obrst *et al.* [43], Duque-Ramos *et al.*, Tartir et al. [54] and Brank *et al.* [6]. From this taxonomy, we evaluate our ontology by addressing a representative range of competency questions that illustrate the task-based and application-based capabilities of the EO. We also evaluate the EO by applying the evolution-based technique proposed by Tartir *et al.* [54] and analyzing the capabilities introduced by version 2.0 of our ontology.

We evaluate the task-based and application-based abilities of the EO to assist system designers in providing support and include user requirements, address explanation dependent questions across the illustrated use cases. In Table 9 and 10, we present the competency questions that we have developed to evaluate the task-based and application-based capabilities of the EO, respectively. In each of these tables, we show the setting for the competency question related to the question and its answer. These competency questions are realized via SPARQL queries that are run on the EO or its companion use case KGs (Section 4). These SPARQL queries can be browsed through our resource website (Section 5.1). Additionally, as part of the answers, we also include additional metrics in both Table 9 and 10, to help assess the complexity of addressing the competency questions and we borrow these metrics from Kendall and McGuinness's recent Ontology Engineering [33] book. These metrics include the *overall query length for addressing the competency question*, *were any property restrictions accessed in retrieving the answer*, *did a reasoner need to be run for the result* and *finally were there any filter clauses required to narrow down the result*.

6.1. Evolution-based evaluation

We also evaluate the additions to the EO model since its first iteration described in Chari *et al.* [13] using the evolution-based evaluation method mentioned in Muhammad *et al.*'s taxonomy [1]. However, as analyzed by Muhammad *et al.*, it is hard to quantify the evolution-based evaluation technique of Tartir *et al.* [54] for knowledge gain provided by the updates to the ontology model. From a qualitative assessment, we find that the additions to the EO model helped us better represent capabilities including:

- Capture more granular representations of 'AI methods' and their interactions with the explanation types, and support more ways to generate explanations.
- Introduce characteristics at various strategic attributes that contribute to explanations (e.g., at the system, user, and object classes), which provide the flexibility to define characteristics at multiple levels and allow for better considering explanation types through the restriction of equivalent classes. For example, we could better represent restrictions against the 'contextual knowledge' class with the broader characteristic scope, and therefore, more patterns can be considered as matches for this class, which is a primary contributor to the 'contextual explanation' type.
- Include more of the contributing attributes of the explanation ecosystem itself (such as capturing the 'system' in which the 'AI methods' are run), which helps maintain better provenance of the infrastructure contributing to the explanations.

Table 9

A catalog of competency questions and candidate answers produced by our EO

Setting	Competency Question	Answer	SPARQL Query length	Property Restrictions accessed?	Inference Required?	Filter Statements
System Design	Q1. Which AI model(s) is/are capable of generating this explanation type (e.g. trace-based)?	Knowledge-based systems, Machine learning model: decision trees	8	Yes	No	No
System Design	Q2. What example questions have been identified for counterfactual explanations?	What other factors about the patient does the system know of? What if the major problem was a fasting plasma glucose?	4	No	No	No
System Design	Q3. What are the components of a scientific explanation?	Generated by an AI Task, Based on recommendation, and based on evidence from study or basis from scientific method	2	Yes	No	No
System Analysis	Q4. Given the system has ranked specific recommendations by comparing different medications, what explanations can be provided for that recommendation?	Contrastive explanation	8	Yes	No	No
System Analysis	Q5. Which explanation type best suits the user question asking about numerical evidence, and how does a system generate such an answer?	Explanation type: statistical; System: run ‘Inductive’ AI task with ‘Clustering’ method to generate numerical evidence	18	Yes	No	No
System Analysis	Q6. What is the context for data collection and application of the contextual explanation, say for example from the health survey analysis use case?	Explanation type: contextual Environmental context: ‘Early childhood questionnaire’ in a US location	5	No	Yes	No

6.2. *Task-based evaluation*

With the increasing demand to support explainability as a feature of user-facing applications, thus improving uptake and usability of AI and ML-enabled applications [11,18,20,27], it is crucial for system designers to understand how to support the kinds of explanations needed to address end user needs. An evolving landscape of the explanations, goals, and methods that support them, complicates the task, but querying the EO can help answer such questions in a standalone format. For the task-based abilities, we aim to showcase how the EO can provide “human ability to formulate queries using the query language provided by the ontology” [43] and “the degree of explanation capability offered by the system, the coverage of the ontology in terms of the degree of reuse across domains” [43].

We detail some of the support that the EO can provide to help system designers understand the main entities interacting with explanations in Table 9. The support that we illustrate includes querying the ‘AI Method’ and ‘AI Task’ that generates the explanation, the example questions that different explanation types can address, and the more nuanced parts of explanation types, such as their components and when they can be generated. The table also shows a set of competency questions and answers that are retrieved from the EO. For answers in this table, we use, for better understanding, simpler descriptions than the results returned by the SPARQL query. The full set of results

Table 10

Example questions that can be asked of our use case knowledge graphs that are modeled using our EO

Use Case	Competency Question	Answer	SPARQL Query Length	Property Restrictions Accessed?	Inference Required?	Filter Statements
Food Recommendation	Q1. What explanation types are supported?	Contextual and Contrastive	2	No	Yes	No
Food Recommendation	Q2. Why should I eat spiced cauliflower soup?	Cauliflower is in season.	5	No	Yes	Yes
Proactive Retention	Q2. What is the retention action outcome for employee 1?	Employee 1 is likely to remain in the same organization.	4	No	No	Yes
Health Survey Analysis	Q3. Who are the most representative patients in the income questionnaire?	Patient 1 and 2.	3	No	Yes	No
Health Survey Analysis	Q5. Which questionnaire did patient 1 answer?	Income, early childhood and social determiners.	3	No	Yes	Yes
Medical Expenditure	Q6.What are the rules for high-cost expenditure?	Individuals are in poor health, have limitations in physical functioning and are on health insurance coverage.	4	No	Yes	Yes
Credit Approval	Q7. What factors contribute most to a loan applicants credit approval?	Facts: Number of satisfactory trades and risk estimate value	7	No	Yes	No

can be browsed through our resource website.[12]

We split the questions across two settings, including during ***system design*** (questions 1–3 in Table 9) when a system designer is planning for what explanation methods and types of support are needed, based on the user and business requirements. The other setting, during ***system analysis*** (questions 4–6 in Table 9), when they are trying to understand what explanation types can be supported given system outputs at their disposal and/or the dependencies of the explanation type instances in their use case KGs on its attributes, such as the system state and context.

Delving further into answers for system design questions from Table 9 such as ‘What particular explanation type addresses a prototypical question?’ can signal to system designers what explanation types can best address the user questions in their use case. Additionally, knowledge of what components populate certain explanation types (Q1. and Q3.) can help system designers plan for what inputs they would need to support these explanation types in their use case.

Similarly, in system analysis settings, which could include, once the system designer knows what system outputs they can build explanations of (Q4–5.), or once a reasoner has been run on the use case KG, or once the explanations inferred via the EO have been displayed (Q6.), a system designer might need to be ready for additional questions that users or system developers might have. These include questions such as ‘What is the context for the data populating the explanation?’ or ‘In what system setting did this explanation get populated?’ Upon querying the EO or use case KGs, we can support answers to such system analysis questions, giving system designers more insights around explanations supported in or to support in their use cases. Besides the general questions about the explanations, a system designer might need to assist interface designers or users with more domain-specific questions about the explanation instances. We can support these through the domain-level instantiations that we allow for in the use case KGs. Examples of these application-based evaluations can be seen in Table 10 and are described in the next section.

In summary, the competency questions we address in Table 9 provide examples for a task-based evaluation of the EO as a model to support user-centered explanations and we checked the utility of these questions against a small expert panel of system designers from our lab. We report their insights in the discussion section (Section 8). We are

[12] SPARQL query results: https://tetherless-world.github.io/explanation-ontology/competencyquestions/.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX eo: <https://purl.org/heals/eo#>
PREFIX ep: <http://linkedu.eu/dedalo/explanationPattern.owl#>
PREFIX sio: <http://semanticscience.org/resource/>

SELECT ?subject ?data
  WHERE {
?subject a eo:DataExplanation .
?subject rdfs:label ?sl .
?subject ep:isBasedOn ?o .
?o sio:SIO_001277 ?data .
  filter( regex(str(?sl), "high-cost", "i" ))

}
```

Listing 3. A SPARQL query run on the medical expenditure knowledge graph, that retrieves the rules associated with a data explanation for high-cost expenditure to answer a competency question of the kind, "What are the rules for high-cost expenditure?

soliciting additional suggestions for more types of questions that showcase a broader range of the EO's capabilities and suggestions can be submitted via recommendations on our website.[13]

6.3. Application-based evaluation

When system designers represent use case specific content, they often need to communicate these representations to other teams, including interface designers who support these use case KGs on UIs, or system developers who want to ensure that they correctly capture system outputs. Hence, in such scenarios, the system designers would need to provide results concerning the domain-specific content in their use case KGs, which could span questions like "What entities are contained in the contrastive explanation?" to more specific questions about particular objects in the use case, such as "What is the outcome for employee 1?" Through the EO, we enable the system designer to provide application-specific details about the explanations supported in their use cases, that improve the presentation of the "output of the applications" [6].

Some examples of questions that we can handle for the use cases we support from Section 4, are shown in Table 10. In these instances from Table 10, we query the use case KGs for domain-specific content by leveraging the properties of the EO model, as defined between the entities that contribute to explanation instances in these KGs.

Some examples of domain-specific content that can be queried (see Table 10) include questions like, 'what are the facts contributing to a contrastive explanation (Q7)', 'what is the 'system recommendation' linked to employee 1 (Q3)', and 'what are some rules for trace-based explanations in the KG (Q6)?' Answers to such questions can provide insights on the entities contributing to explanations. More specifically, the answers to questions in Table 10, can help system designers convey to interface designers what entities can be shown on the UI concerning these explanations (Q1, Q3–Q4) or even help application users and system developers navigate the explanation dependencies to understand the interactions between the entities contributing to these user-centered explanations (Q2, Q5–Q7). An example of a SPARQL query that implements "Q6. What are the rules for high-cost patient expenditure?" from Table 10, can be viewed in Listing 3 and the results can be seen in Table 11.

In summary, representing explanations via the EO model allows the content supporting the explanations to be queried easily and through multiple depths of supporting provenance. This is also a step toward supporting an interactive design for explanations, such as the one mentioned in Lakkaraju et al. [34]. We aim for these example questions addressed through our use case KGs, to provide guidance to system designers on the types of questions that they could support in their own use case KGs.

[13] Call for Participation: https://tetherless-world.github.io/explanation-ontology/competencyquestions/#call.

Table 11

Results of the SPARQL query to retrieve the higher-degree rules associated with a data explanation instance from the medical expenditure use case

Explanation	Rule
Explanation based on high-cost patient pattern 2.	Self-reported poor health – true
Explanation based on high-cost patient pattern 2.	Limitations in physical functioning – present
Explanation based on high-cost patient pattern 2.	Health insurance coverage – present

7. Related work

The increased awareness on AI explainability in the recent years has resulted in the publication of several papers. Some provide solutions to explain decisions and behavior of ML methods [2,3,42]. Others are position statements and survey papers pointing to the need for user-driven and user-centered explainability that takes into account the requirements of the system designers and end-users during the explanation composition and generation process [11,21,27,34]. Furthermore, several publications find that model explanations alone, which are often either scores or model outputs that are not comprehensible to users who lack the knowledge of system and methods development [60], cannot answer the diverse set of questions that users have for the explainability needs of AI models [28,34,36]. Hence, for user consumption, explanations need to be grounded in additional supporting data, context, and knowledge [11,12]. Some of the reasons why we posit and others find that model explanations or a single set of explanations are insufficient are that when humans are looking to trust an AI system, they are seeking support that they are familiar with [28,34] that could include explanations that answer questions that appeal to their reasoning process, are expressed in the domain knowledge that they are accustomed to, and provide additional information to understand the AI system and its workings. Hence, these requirements for user-centered explanations can often not be satisfied by a single explainability method or question type and require different explanation types to be presented to humans that can help them reason through the AI model's decision and its explanations via different paths.

Ontologies and KGs capture an encoding of associations between entities and relationships in domains, and hence, can be used to inform upstream tasks [24,47], guide/constrain ML models [5], and structure content for the purpose of organization [39,58]. User-centered explanations are composed of different components, such as outputs of AI/ML methods and prior knowledge, and are also populated by content annotated by different domain ontologies and KGs (of which there are many). The former proposition of composing explanations from components has been attempted less frequently [13,16,57,58] and at different degrees of content abstraction, hence providing open challenges to represent explanations semantically. Additionally, there have been two multidisciplinary, comprehensive, and promising reviews [35,59] highlighting the applications of KGs to explainable AI, either solely as the data store to populate explanations, or as aids to explain AI decisions from the knowledge captured by the KG encodings. Efforts to use ontologies and KGs to improve the explainability of AI models will become increasingly popular since several publications point out that single scores from ML models are hard for subject matter experts (SMEs) to interpret directly [9,26,46]. Here, we review semantic efforts to represent explanations and highlight how the EO is different in terms of its overall goal and representation.

One of the early efforts of an ontology to represent explanations was by Tiddi et al. [58], who designed the Explanation Patterns (EP) ontology to model attributes of explanations from a philosophy perspective of explanation and its dependencies on what phenomenon generated the explanation and what events they are based on. They based their ontology design on ontological realism [53], i.e., building their ontology to be as close to how it is defined in theory. The primary use of the EP ontology was to define explanations in multi-disciplinary domains, such as cognitive science, neuroscience, philosophy, and computer science, with a simple general model. The authors mention how the EP model can be applied in conjunction with a graph traversal algorithm like Dedalo [57] to find three components of an explanation, including the **A**ntecedent event (A), the **T**heory they are based on (T), and the **C**ontext in which they are occurring (C), in KGs. However, we found that for the user-centered explanation types (Table 3), the (A, T, C) is often not sufficient when some explanations either do not need all the (A, T, C) components or when other explanations require more than these components. For example, (A, T, C) components are not sufficient in *case-based explanations*, where what cases the explanation is dependent upon needs to be modeled. Additionally, we found that this simplicity was insufficient to support explanations generated by AI methods. Therefore, we added additional classes to support the dependencies of explanations on the methods that produce them and the users that consume them. However, in the spirit of reusing existing ontologies, we leverage certain classes and properties of the EP ontology in our EO ontology model as described in Section 3.1.

In a recent paper, Dalvi and Jansen et al. [16] released an ENTAILMENTBANK for explanations and applied natural-language processing methods to identify trees for the facts that are most relevant to a Question-Answer (QA) pair. While they are trying to identify entailments, or the facts that are most pertinent to an answer in a QA setting, several publications [18,28,34] find that domain experts are often aware of the base knowledge that support explanations and do not always appreciate the additional theory. In the EO, we model a comprehensive set of literature-derived explanation types that can address various questions by system designers. These explanation types can provide users the varied support [28,34] they seek in terms of information to help them better understand the AI methods output. For example, contextual explanations situate answers and scientific explanations provide evidence to reason about the supporting literature. As for composing the explanations, we describe how the description of sufficiency conditions on these explanation types allows them to be built from KGs (Section 3.2). In a similar vein, Teze et al. [55] present style templates to combine assertional, terminological and ontology terms to support seven different user-centered explanation types including statistical, contextual, data-driven, simulation-based, justification-based, contrastive and counterfactual. However, their work doesn't use traditional ontology languages like RDF and OWL and is therefore less interoperable with standard semantic frameworks. Additionally, they focus more on integrating the outputs of logical reasoners to populate explanation types, instead of a wider breadth of AI explanation methods, such as those listed in Zhou et al. [64] and Arya et al. [2,3]. In the EO, we design a simple, yet comprehensive, model to represent user-centered explanation types that account for their generational needs and impact on the user.

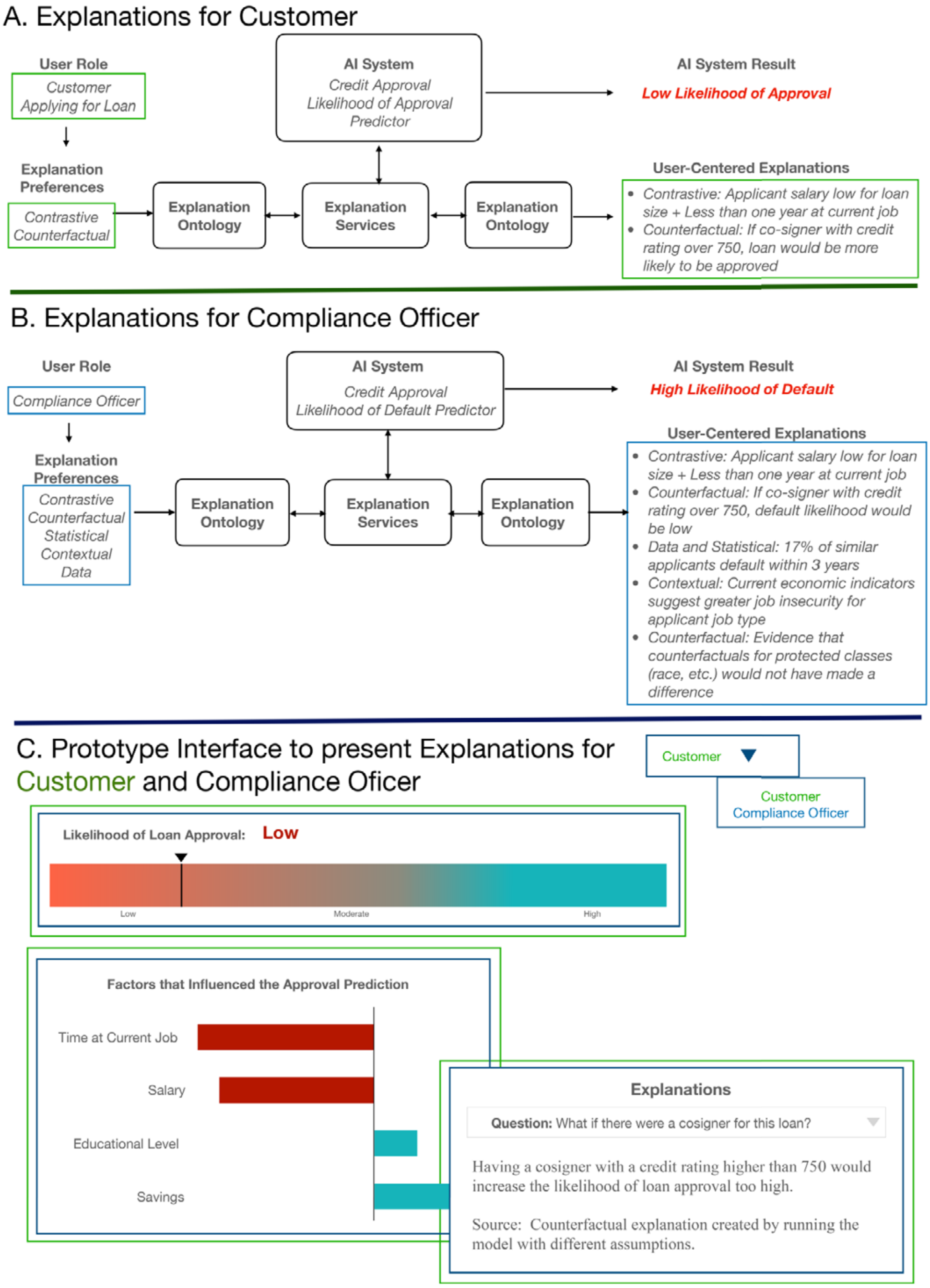
8. Discussion

Here we discuss different aspects of the EO, including desired features and design choices, the relevance of results and limitations, and future outlook.

How is EO considered to be general-purpose? We have described a general-purpose and mid-level ontology, the EO, that can be leveraged to represent user-centered explanations in domain applications. In the EO, we encode the attributes that contribute to explanations in a semantic representation as an ontology, and by doing so, the EO can be used to structure explanations based on linkages to the needs the explanations support and the method chains that generate them (Section 3). The ability to support different explanation types enables the EO to be a tool that can provide users with different views required to reason over the recommendations provided by AI-enabled systems. In addition to explainability being diverse in terms of the different types of needs that users seek from explanations (Fig. 10), explanations are also domain-specific and are driven by the system outputs and domain knowledge in the application domain or use case [10,34]. Hence, although the EO is a mid-level ontology, it needs to be broad enough to support the representation of explanations across domains. Through our use case KGs (Section 4), we have demonstrated how system designers, as our intended users of the EO, can root the domain-specific concepts in our EO model and, upon running a reasoner on their KGs, classify their representations into the user-centered explanation types we support in the EO.

What features were introduced in EO 2.0 and why? This paper is a significantly expanded version of an earlier conference paper on the EO [13], and here we also introduce new use cases and explanation types. With the introduction of the six new explanation types from Zhou et al. [64], we can now infer a broader range of user-centered explanation types in use cases. For example, we describe here how to infer data and rational explanations in the proactive retention, credit approval, and medical expenditure use cases. Also, without expanding the explanation method trees in the EO, we would not have been able to adequately capture the outputs from explanation methods in the AIX-360 use cases. Hence, our additions to the EO model introduced in this paper can provide expanded expressivity, leading to better applicability with the current explainable AI landscape.

What are design choices made during EO development and how do they apply? The EO itself is lightweight, and we have imported two standard scientific ontologies, SIO-O and Prov-O. We reuse a lot of classes and we only


A. Explanations for Customer
User Role
Customer Applying for Loan
Explanation Preferences
Contrastive Counterfactual
AI System
Credit Approval Likelihood of Approval Predictor
Explanation Ontology
Explanation Services
Explanation Ontology
AI System Result
Low Likelihood of Approval
User-Centered Explanations
B. Explanations for Compliance Officer
User Role
Compliance Officer
Explanation Preferences
Contrastive Counterfactual Statistical Contextual Data
AI System
Credit Approval Likelihood of Default Predictor
Explanation Ontology
Explanation Services
Explanation Ontology
AI System Result
High Likelihood of Default
User-Centered Explanations
C. Prototype Interface to present Explanations for Customer and Compliance Oficer
Customer
Customer
Compliance Officer
Likelihood of Loan Approval: Low
Low
Moderate
High
Factors that Influenced the Approval Prediction
Time at Current Job
Salary
Educational Level
Savings
Explanations
Question: What if there were a cosigner for this loan?

- Contrastive: Applicant salary low for loan size + Less than one year at current job
- Counterfactual: If co-signer with credit rating over 750, loan would be more likely to be approved

- Contrastive: Applicant salary low for loan size + Less than one year at current job
- Counterfactual: If co-signer with credit rating over 750, default likelihood would be low
- Data and Statistical: 17% of similar applicants default within 3 years
- Contextual: Current economic indicators suggest greater job insecurity for applicant job type
- Counterfactual: Evidence that counterfactuals for protected classes (race, etc.) would not have made a difference

Having a cosigner with a credit rating higher than 750 would increase the likelihood of loan approval too high.

Source: Counterfactual explanation created by running the model with different assumptions.

Fig. 10. Illustrating the varying preferences that users have for explanations in a credit approval use case. This illustration shows that explanations can be populated differently for user groups, such as the customer (A).) and compliance officer (B.). The Explanation Ontology (EO) can be used in both user scenarios A). and B). to support the composition of explanations from individual data sources and methods. Further, structuring explanations across user groups using common templates provided in the EO, can help them be rendered flexibly on an interface such as the example in C).

introduce classes and properties that do not exist currently to represent explanations. This design choice of reuse is reflected in the ontology metrics reported in Table 2 and is a feature which can ensure interoperability of the EO with other ontologies that also use the standard scientific ontologies that we reuse. From a modeling perspective, in the EO, we are aiming for expressivity in that we capture multiple paths to compose explanation types which contribute to a somewhat slow reasoner performance. Since the reasoner is typically run only once in a use case, or it is a process that is not run in real-time, the speed limitation might not be an issue when displaying such pre-computed results.

What is the coverage of the EO and how can the ontology be adapted? The EO provides an approximate representation of content in the evolving literature surrounding XAI. From a modeling perspective, we have modeled attributes of explanations that we deemed essential to represent explanations that address the standard set of question templates identified in Vera et al. [36] (e.g., Why, Why not, What, How, etc.), depending on a set of models, knowledge, and data resources, as described in computer science and explanation sciences literature [20,27,40]. We also modeled attributes required to represent the system interface user attributes, as described in the human-interaction literature [17,37,63]. Interested users can also extend EO with additional classes as they deem necessary, by referring to the descriptions of the EO model available in this paper and on our resource website. Additionally, we update the EO model often, ensuring that it is current. In terms of adding more granular and indirectly connected attributes to explanations, we are also investigating how to represent system attributes, such as error traces, metrics and method parameters, and to link such details in user-centered explanation types that would provide value to a large range of domain and non-domain users.

Discussion of Results: We have evaluated the capabilities of the EO in assisting system designers, our intended users, both from a task perspective of the planning process in supporting explanations in their use cases and addressing domain-specific questions that might arise around their use cases. Hence, we use the task-based and application-based evaluation techniques presented in Muhammad et al.'s well-curated taxonomy of ontology evaluation methods [1]. We crafted these competency questions borrowing from our expertise in the explainable AI domain, i.e., from our previous experiences of interactions with end-users where they have alluded to specific needs for explainability [10,13,28], and from literature reviews of the kinds of questions that users want addressed from explanations [36] and what explanation taxonomies cover [2,3]. We also walked a small expert panel of two system designers in our lab through our evaluation approach and presented them with probing questions like do the competency questions serve their needs when they are trying to use the EO and what other questions they would like to see addressed. They made some suggestions around rewording some task-based questions 9 to be more clear and expressed interest in seeing the domain capabilities of the EO (Table 10). We have made changes to the evaluation to reflect the expert panel suggestions. While our evaluation was not designed to be exhaustive, it is representative of capabilities that the EO can enable around making explanations composable from its dependencies and allowing explanations to be probed to support further user-driven questions around them.

Further, the values for different metrics, including query length, property restriction, and whether or not additional mechanisms like inference are required and filter statements need to be applied, that are reported against queries for each competency question in Table 9 and Table 10, reflect the design choices that we have made in the EO. To elaborate, queries in the task-based evaluation have longer query lengths since they access property restrictions defined against explanation types supported in the EO to answer questions around the components of these explanation types. On the other hand, since the EO provides capabilities of explanations to be classified into explanation types, queries around domain-specific content of inferred explanations in the exemplar use case KGs have shorter query lengths. While interested system designers could use the queries we have provided in this paper (Section 6) as a reference for other queries they might have around the EO or want addressed by the EO, we also have an active developer community that can assist in crafting these queries. We have a call for participation on our website for interested users.[14]

Future Outlook: Overall, the EO can be thought of as a semantic representation that allows for easy slot-filling of user-centered explanation types in terms familiar to most system designers. In the future, we hope to develop a natural-language processing method that would interface with the explanation slots in the EO's use case KGs to build

[14] Competency question support: https://tetherless-world.github.io/explanation-ontology/competencyquestions/#call.

natural-language explanations. We continue to provide and update our open-source documentation for using the EO model to support user-centered explanation types in use cases that span various domains. The EO is a solution to combine data, knowledge and model-capabilities to compose user-centered explanations that can address a wide range of user questions and provide multiple views and thus support human reasoning of AI outputs [28,34] (as illustrated in the example in Fig. 10).

9. Conclusion

We have presented a significantly expanded explanation ontology that can serve as a resource for composing explanations from contributing components of the system, interface, and user- attributes. We have modeled the mid-level ontology to be used as a cross-domain resource to represent user-centered explanations in various use cases. In addition, within the ontology, we model fifteen literature-derived, user-centered explanation types and define equivalent class restrictions against these types that allow for explanations to be classified into these patterns. In this paper, we have provided guidance for a system designer, our intended user, to apply our ontology in their use cases. This guidance includes descriptions of five open-source use cases that use our ontology, and answers to competency questions that demonstrate the EO's task and application-based capabilities. We aim for the competency questions to serve as a means for system designers to familiarize themselves with the capabilities of our ontology, to support explanations and understand what types of content-specific questions can be asked around explanations in their use cases. Finally, we hope for this open-sourced ontology to serve as a resource to represent user-centered explanations in various use cases and allow for these explanations to be supported by a broad range of AI methods and knowledge sources while accounting for user requirements.

Acknowledgements

This work is done as part of the HEALS project, and is partially supported by IBM Research AI through the AI Horizons Network. We thank our colleague from RPI, Rebecca Cowan, Henrique Santos and Jamie McCusker, who provided insight and expertise in the writing of this paper.

References

[1] M. Amith, Z. He, J. Bian, J.A. Lossio-Ventura and C. Tao, Assessing the practice of biomedical ontology evaluation: Gaps and opportunities, *Journal of biomedical informatics* **80** (2018), 1–13. doi:10.1016/j.jbi.2018.02.010.

[2] V. Arya, R.K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S.C. Hoffman, S. Houde, Q.V. Liao, R. Luss, A. Mojsilović et al., AI explainability 360: Impact and design, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, 2022, pp. 12651–12657.

[3] V. Arya, R.K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S.C. Hoffman, S. Houde, Q.V. Liao, R. Luss, A. Mojsilović et al., One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques, 2019, arXiv preprint arXiv:1909.03012.

[4] D. Bau, B. Zhou, A. Khosla, A. Oliva and A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 6541–6549.

[5] K. Beckh, S. Müller, M. Jakobs, V. Toborek, H. Tan, R. Fischer, P. Welke, S. Houben and L. von Rueden, Explainable Machine Learning with Prior Knowledge: An Overview, 2021, arXiv preprint arXiv:2105.10172.

[6] J. Brank, M. Grobelnik and D. Mladenic, A survey of ontology evaluation techniques, in: *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, Citeseer, Ljubljana Slovenia, 2005, pp. 166–170.

[7] Cancer Biomedical Informatics Grid, Unified Medical Language System, National Cancer Institute Thesarus (NCIT).

[8] Center for Disease Control and Prevention (CDC), National Health and Nutrition Examination Survey.

[9] D.W. Challener, L.J. Prokop and O. Abu-Saleh, The proliferation of reports on clinical scoring systems: Issues about uptake and clinical utility, *Jama* **321**(24) (2019), 2405–2406. doi:10.1001/jama.2019.5284.

[10] S. Chari, P. Chakraborty, M. Ghalwash, O. Seneviratne, E.K. Eyigoz, D.M. Gruen, F.S. Saiz, C.-H. Chen, P.M. Rojas and D.L. McGuinness, Leveraging Clinical Context for User-Centered Explainability: A Diabetes Use Case, 2021, arXiv preprint arXiv:2107.02359.

[11] S. Chari, D.M. Gruen, O. Seneviratne and D.L. McGuinness, Directions for explainable knowledge-enabled systems, in: *Knowledge Graphs for eXplainable AI – Foundations, Applications and Challenges*, Studies on the Semantic Web, I. Tiddi, F. Lecue and P. Hitzler, eds, IOS Press, 2020, to appear.

[12] S. Chari, D.M. Gruen, O. Seneviratne and D.L. McGuinness, Foundations of explainable knowledge-enabled systems, in: *Knowledge Graphs for eXplainable AI – Foundations, Applications and Challenges*, Studies on the Semantic Web, I. Tiddi, F. Lecue and P. Hitzler, eds, IOS Press, 2020, to appear.

[13] S. Chari, O. Seneviratne, D.M. Gruen, M.A. Foreman, A.K. Das and D.L. McGuinness, Explanation ontology: A model of explanations for user-centered ai, in: *International Semantic Web Conference*, Springer, 2020, pp. 228–243.

[14] Y. Chen, A. Subburathinam, C.-H. Chen and M.J. Zaki, Personalized food recommendation as constrained question answering over a large-scale food knowledge graph, in: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 544–552. doi:10.1145/3437963.3441816.

[15] M. Courtot, F. Gibson, A.L. Lister, J. Malone, D. Schober, R.R. Brinkman and A. Ruttenberg, MIREOT: The minimum information to reference an external ontology term, *Applied Ontology* **6**(1) (2011), 23–33. doi:10.3233/AO-2011-0087.

[16] B. Dalvi, P. Jansen, O. Tafjord, Z. Xie, H. Smith, L. Pipatanangkura and P. Clark, Explaining Answers with Entailment Trees, 2021, arXiv preprint arXiv:2104.08661.

[17] A.K. Dey, G.D. Abowd and A. Wood, CyberDesk: A framework for providing self-integrating context-aware services, *Knowledge-based systems* **11**(1) (1998), 3–13. doi:10.1016/S0950-7051(98)00053-7.

[18] S. Dey, P. Chakraborty, B. Chul Kwon, A. Dhurandhar, M. Ghalwash, F.J. Suarez Saiz, K. Ng, D. Sow, K.R. Varshney and P. Meyer, Human-Centered Explainability for Life Sciences, Healthcare and Medical Informatics, *Patterns* (2022).

[19] S. Dey, P. Chakraborty, B.C. Kwon, A. Dhurandhar, M. Ghalwash, F.J.S. Saiz, K. Ng, D. Sow, K.R. Varshney and P. Meyer, Human-centered explainability for life sciences, healthcare, and medical informatics, *Patterns* **3**(5) (2022), 100493. doi:10.1016/j.patter.2022.100493.

[20] F. Doshi-Velez and B. Kim, Towards a rigorous science of interpretable machine learning, 2017, arXiv preprint arXiv:1702.08608.

[21] F. Doshi-Velez, M. Kortz, R. Budish, C. Bavitz, S. Gershman, D. O'Brien, S. Schieber, J. Waldo, D. Weinberger and A. Wood, Accountability of AI under the law: The role of explanation, 2017, arXiv preprint arXiv:1711.01134.

[22] M. Dumontier, C.J. Baker, J. Baran, A. Callahan, L. Chepelev, J. Cruz-Toledo, N.R. Del Rio, G. Duck, L.I. Furlong, N. Keath et al., The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery, *J. Biomed. Semantics* **5**(1) (2014), 14. doi:10.1186/2041-1480-5-14.

[23] FICO Community, Explainable Machine Learning Challenge, MITr.

[24] M. Gaur, K. Faldu and A. Sheth, Semantics of the Black-Box: Can knowledge graphs help make deep learning systems more interpretable and explainable?, *IEEE Internet Computing* **25**(1) (2021), 51–59. doi:10.1109/MIC.2020.3031769.

[25] M. Ghassemi, L. Oakden-Rayner and A.L. Beam, The false hope of current approaches to explainable artificial intelligence in health care, *The Lancet Digital Health* **3**(11) (2021), e745–e750, https://www.sciencedirect.com/science/article/pii/S2589750021002089. doi:10.1016/S2589-7500(21)00208-9.

[26] M. Ghassemi, L. Oakden-Rayner and A.L. Beam, The false hope of current approaches to explainable artificial intelligence in health care, *The Lancet Digital Health* **3**(11) (2021), e745–e750. doi:10.1016/S2589-7500(21)00208-9.

[27] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter and L. Kagal, Explaining explanations: An approach to evaluating interpretability of machine learning, 2018, arXiv preprint arXiv:1806.00069.

[28] D.M. Gruen, S. Chari, M.A. Foreman, O. Seneviratne, R. Richesson, A.K. Das and D.L. McGuinness, Designing for AI explainability in clinical context, in: *Trustworthy AI for Healthcare Workshop at AAAI 2021*, 2021.

[29] K.S. Gurumoorthy, A. Dhurandhar, G. Cecchi and C. Aggarwal, Efficient data representation by selecting prototypes with importance weights, in: *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2019, pp. 260–269. doi:10.1109/ICDM.2019.00036.

[30] S. Haussmann, O. Seneviratne, Y. Chen, Y. Ne'eman, J. Codella, C.-H. Chen, D.L. McGuinness and M.J. Zaki, FoodKG: A semantics-driven knowledge graph for food recommendation, in: *International Semantic Web Conference*, Springer, 2019, pp. 146–162.

[31] M. Hind, D. Wei, M. Campbell, N.C. Codella, A. Dhurandhar, A. Mojsilović, K.N. Ramamurthy and K.R. Varshney, TED: Teaching AI to explain its decisions, in: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 123–129. doi:10.1145/3306618.3314273.

[32] IBM Research Trusted AI, AI Explainability 360 Open Source Toolkit.

[33] E.F. Kendall and D.L. McGuinness, Ontology engineering, *Synthesis Lectures on The Semantic Web: Theory and Technology* **9**(1) (2019), i–102. doi:10.1007/978-3-031-79486-5.

[34] H. Lakkaraju, D. Slack, Y. Chen, C. Tan and S. Singh, Rethinking Explainability as a Dialogue: A Practitioner's Perspective, 2022, arXiv preprint arXiv:2202.01875.

[35] F. Lecue, On the role of knowledge graphs in explainable AI, *Semantic Web* **11**(1) (2020), 41–51. doi:10.3233/SW-190374.

[36] Q.V. Liao, D. Gruen and S. Miller, Questioning the AI: Informing design practices for explainable AI user experiences, in: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–15.

[37] B.Y. Lim, A.K. Dey and D. Avrahami, Why and why not explanations improve the intelligibility of context-aware intelligent systems, in: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ACM, 2009, pp. 2119–2128. doi:10.1145/1518701.1519023.

[38] S.M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, *Advances in neural information processing systems* **30** (2017).

[39] D.L. McGuinness and P.P. Da Silva, Explaining answers from the semantic web: The inference web approach, *Web Semantics: Sci., Services and Agents on the World Wide Web* **1**(4) (2004), 397–413. doi:10.1016/j.websem.2004.06.002.

[40] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, *Artificial Intelligence* **267** (2019), 1–38. doi:10.1016/j.artint.2018.07.007.

[41] B. Mittelstadt, C. Russell and S. Wachter, Explaining explanations in AI, in: *Proc. of the Conf. on Fairness, Accountability, and Transparency*, ACM, 2019, pp. 279–288. doi:10.1145/3287560.3287574.

[42] I. Nunes and D. Jannach, A systematic review and taxonomy of explanations in decision support and recommender systems, *User Modeling and User-Adapted Interaction* **27** (2017), 393–444. doi:10.1007/s11257-017-9195-0.

[43] L. Obrst, W. Ceusters, I. Mani, S. Ray and B. Smith, The evaluation of ontologies, in: *Semantic Web*, Springer, 2007, pp. 139–158. doi:10.1007/978-0-387-48438-9_8.

[44] Ontograf – Protege Wiki, Accessed: 2022-04-04.

[45] I. Padhiar, O. Seneviratne, S. Chari, D. Gruen and D.L. McGuinness, Semantic modeling for food recommendation explanations, in: *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, IEEE, 2021, pp. 13–19. doi:10.1109/ICDEW53142.2021.00010.

[46] Y. Park, G.P. Jackson, M.A. Foreman, D. Gruen, J. Hu and A.K. Das, Evaluating artificial intelligence in medicine: Phases of clinical research, *JAMIA open* **3**(3) (2020), 326–331. doi:10.1093/jamiaopen/ooaa033.

[47] A. Raghu, J. Guttag, K. Young, E. Pomerantsev, A.V. Dalca and C.M. Stultz, Learning to predict with supporting evidence: Applications to clinical risk prediction, in: *CHIL'21: Proceedings of the Conference on Health, Inference, and Learning*, ACM, 2021, pp. 95–104. doi:10.1145/3450439.3451869.

[48] M.T. Ribeiro, S. Singh and C. Guestrin, Why should I trust you?: Explaining the predictions of any classifier, in: *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1135–1144.

[49] S.C. for Biomedical Research (BMIR), Protege.

[50] S.C. for Biomedical Research (BMIR), Ontology Metrics.

[51] A.A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne and E. Motta, The computer science ontology: A large-scale taxonomy of research areas, in: *International Semantic Web Conference*, Springer, 2018, pp. 187–205.

[52] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L.J. Goldberg, K. Eilbeck, A. Ireland, C.J. Mungall et al., The OBO foundry: Coordinated evolution of ontologies to support biomedical data integration, *Nature biotechnology* **25**(11) (2007), 1251–1255. doi:10.1038/nbt1346.

[53] B. Smith and W. Ceusters, Ontological realism: A methodology for coordinated evolution of scientific ontologies, *Applied ontology* **5**(3–4) (2010), 139–188. doi:10.3233/AO-2010-0079.

[54] S. Tartir, I.B. Arpinar and A.P. Sheth, Ontological evaluation and validation, in: *Theory and Applications of Ontology: Computer Applications*, Springer, 2010, pp. 115–130. doi:10.1007/978-90-481-8847-5_5.

[55] J.C.L. Teze, J.N. Paredes, M.V. Martinez and G.I. Simari, Engineering User-centered Explanations to Query Answers in Ontology-driven Socio-technical Systems, Under Review, *Semantic Web Journal* (2022).

[56] The Apache Software Foundation, Vol. 2.0 License.

[57] I. Tiddi, M. d'Aquin and E. Motta, Dedalo: Looking for clusters explanations in a labyrinth of linked data, in: *European Semantic Web Conf.*, Springer, 2014, pp. 333–348.

[58] I. Tiddi, M. d'Aquin and E. Motta, An ontology design pattern to define explanations, in: *Proceedings of the 8th Int. Conf. on Knowledge Capture*, 2015, pp. 1–8.

[59] I. Tiddi and S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, *Artificial Intelligence* **302** (2022), 103627. doi:10.1016/j.artint.2021.103627.

[60] S. Tonekaboni, S. Joshi, M.D. McCradden and A. Goldenberg, What clinicians want: Contextualizing explainable machine learning for clinical end use, in: *Machine Learning for Healthcare Conference*, PMLR, 2019, pp. 359–380.

[61] J. van der Waa, M. Robeer, J. van Diggelen, M. Brinkhuis and M. Neerincx, Contrastive explanations with local foil trees, 2018, arXiv preprint arXiv:1806.07470.

[62] S. Wachter, B. Mittelstadt and C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GPDR, *Harv. JL & Tech.* **31** (2017), 841.

[63] D. Wang, Q. Yang, A. Abdul and B.Y. Lim, Designing theory-driven user-centric explainable AI, in: *Proceedings of the 2019 CHI Conf. on Human Factors in Computing Systems*, 2019, pp. 1–15.

[64] J. Zhou, A.H. Gandomi, F. Chen and A. Holzinger, Evaluating the quality of machine learning explanations: A survey on methods and metrics, *Electronics* **10**(5) (2021), 593. doi:10.3390/electronics10050593.